



University of
Zurich^{UZH}

*Burkhard Stiller, Alberto Huertas, Bruno Rodrigues, Chao Feng,
Eryk Schiller, Jan von der Assen, Katharina O. E. Müller,
Nazim Nezhadsistani (Edts)*

Communication Systems XVI

TECHNICAL REPORT – No. IFI-2023.01

June 2023

University of Zurich
Department of Informatics (IFI)
Binzmühlestrasse 14, CH-8050 Zürich, Switzerland



Contents

1 Scalability Challenges in Healthcare Blockchains - An Overview	3
<i>Eyluel Goekce Harputluoglu</i>	
2 Promises of the 5G Network	23
<i>Sabyasachi Banik</i>	
3 SCION: A Clean-slate Internet Architecture	49
<i>Elliott Wallace</i>	
4 ML Models for Textual Data Anonymization in NLP	71
<i>Nicolas Kohler and Ramize Abdili</i>	
5 Security Concerns of Privacy-preserving and Decentralized Artificial Intelligence	97
<i>Huiran Duan, Han Yang</i>	
6 The Impact of Open Source on Cybersecurity	141
<i>Lucas Krauter</i>	
<i>Niklas A. Schmatloch</i>	
9 ChatGPT in Network Security - An Overview of its Capabilities for Good and Bad Use Cases	165
<i>Ahmad Eynawi</i>	
10 Ultra-wideband Radio Technology Today: Infrastructure, Security, and User Trust	199
<i>Mohamed Zahir Mohamed Wazeer</i>	

Chapter 1

Scalability Challenges in Healthcare Blockchains - An Overview

Eyhuel Goekce Harputluoglu

Today, blockchain technology has a growing potential to revolutionize many sectors in the industry, including healthcare by providing a secure and transparent platform for managing medical records, ensuring patient privacy and security, and facilitating clinical research. However, one of the major challenges that need to be mentioned here is the scalability issue in blockchain systems. The size of the blockchain grows rapidly with the increased number of nodes and transactions and this leads to longer transaction times and higher processing costs in the network. In the healthcare sector, blockchain technology can change the way that medical records are stored, accessed, and shared. This newer technology can improve the interoperability of medical records between different healthcare providers, increase patient privacy and security, and facilitate clinical research. Although, the scalability issue is a major concern for healthcare blockchains. As the amount of medical data generated daily increases, the storage and processing requirements for such data become enormously big and the current blockchain technology may not be able to handle it effectively. There are some proposals to various approaches in order to overcome this issues in blockchain systems. This paper focuses on the current scalability challenges and explores different solutions such as on-chain, off-chain, redesigning blockchain as well as storage optimisation. By addressing the scalability issue, the potential benefits of blockchain technology in healthcare can be fully realized, leading to improved patient benefits, reduced costs, and enhanced collaboration between healthcare providers.

Contents

1.1	Introduction	5
1.1.1	Usage of Blockchain in Healthcare Systems	6
1.2	Related Work	6
1.3	Challenges	7
1.3.1	Scability Challenges	7
1.3.2	Current challenges of blockchain and IoT in healthcare	8
1.4	Solutions	13
1.4.1	Off-Chain Approach	13
1.4.2	On-Chain Approach	14
1.4.3	Storage Optimisation	14
1.4.4	Redesigning Blockchain	15
1.5	Conclusion	16
1.6	Appendix I	20

1.1 Introduction

Blockchain technology is a futuristic innovation with many possible applications in various industries such as finance, supply chain management, healthcare and more. Blockchain technology is known as a private, secure, trustworthy, and transparent information exchange performed in a decentralised manner. It is considered as 5th disruptive technology after mainframes, PCs, the internet, mobile communications and social media[1]. In cryptocurrency world, this technology became very popular and has found many takers in various applications in different fields, such as voting, healthcare, predictions, data storage and so on. However, this popularity led to unexpected growth of Blockchain network [1].

It is a decentralized and distributed digital ledger that allows transactions to be recorded securely and transparently without the need for intermediaries or central authorities. The blockchain technology has been proven effective in preventing fraud, improving security, and increasing efficiency, but one of its major challenges is scalability[8].

Scalability refers to the ability of a blockchain network to handle an increasing amount of workload in other words transactions without sacrificing its performance or security[2]. Scalability issues arise due to the limited processing power and storage capacity of the individual nodes in the blockchain network. As the number of nodes and transactions increases, blockchain becomes larger and this eventually causes longer transaction times and higher processing costs. This can significantly affect the usability and adoption of blockchain technology in various industries[4].

In the healthcare industry, blockchain technology has a great potential to transform the way medical records are stored, used, and shared. It can improve the interoperability of medical records between different healthcare providers, increase patient privacy and security, and facilitate clinical research. However, the scalability issue is a concern for healthcare blockchains with the larger medical data which is being generated daily. In this case, the storage and processing requirements are increasing accordingly, and the current blockchain technology may not be able to handle it effectively[17].

To overcome the scalability issue, various approaches have been proposed, such as sharding, sidechains, off-chain solutions etc. where Sharding involves dividing the blockchain network into smaller partitions (shards), that can process transactions independently or Sidechains allow for the creation of separate blockchain networks that can be linked to the main blockchain when needed while Off-chain solutions involve moving some of the transaction processing off the main blockchain to improve performance.

Although there are many benefits of those approaches, they also come with their own set of challenges, such as increase in complexity, reduced security, and a potential for centralization. Therefore, further research and development are required to find a sustainable and secure solution to the scalability issue in healthcare blockchains.

All together, blockchain technology has the possibility to be used in various fields in industry, including healthcare, by improving security, transparency, and efficiency. However, the scalability issue remains a major challenge that needs to be addressed. The

use of advanced technological solutions and collaboration between users from industries, researchers, and regulators are crucial in finding a sustainable solution to the scalability issue in healthcare blockchains.

1.1.1 Usage of Blockchain in Healthcare Systems

Blockchain technology is gaining momentum in healthcare systems as a promising solution for secure, efficient, and decentralized data management. With the increasing amount of medical data generated daily, there is a growing need for secure and inter-operable electronic medical records (EMRs) that can be accessed and shared across different healthcare providers[17]. Blockchain technology has the potential to transform the healthcare industry by providing a secure and transparent platform for managing medical records, ensuring patient privacy and security, and facilitating clinical research [17] [2].

One advantages of blockchain technology in healthcare is its ability to provide secure storage of medical records. Unlike traditional centralized databases where a single point of failure can compromise the entire system, blockchain technology distributes the data across multiple nodes in the network, making it virtually impossible to alter or hack, which ensures that the integrity and confidentiality of medical records are accomplished and this is critical for maintaining patient trust and complying with regulatory requirements.

Additionally, blockchain technology can enhance the interoperability of medical records between different healthcare providers. It gives the opportunity to create a decentralized and standardized platform for sharing medical data. Hence, blockchain can reduce the administrative burden of managing medical records and improve the quality of care. Patients might also benefit from greater control and ownership of their medical data, which can improve their engagement and adherence to treatment plans[20].

Another application where blockchain technology in healthcare can be used is the facilitation of clinical research. By providing a secure and transparent platform for collecting and analyzing medical data, blockchain technology can accelerate the discovery of new treatments and therapies in the health sector. Researchers can also benefit from greater access to medical data which can increase the quality and the reliability of their research findings[20] [2].

Needless to say that there are many benefits of blockchain technology in healthcare, however we face with some challenges that needs to be mentioned, such as scalability, interoperability, and regulatory compliance. With the further research and developments in this areas, it is likely to unlock the full potential of blockchain technology in healthcare.

1.2 Related Work

Scalability is an important issue for the widespread adoption of blockchain technology in the various fields including healthcare. The current limitations of blockchain such as low transaction throughput and high latency, may impede the efficient and timely processing of

large volumes of medical data. This section provides some recent researches on scalability solutions for blockchain in healthcare.

There are some approaches proposed to improve the scalability of blockchain technology in healthcare. One very promising solution is that the usage of off-chain scaling techniques such as payment channels and sidechains. Payment channels allow for a high volume of transactions to be processed off-chain, reducing the load on the blockchain network. Sidechains, on the other hand, allow for the creation of separate blockchain networks that can be linked to the main chain, enabling the parallel processing of transactions[19].

Another approach for improving scalability in networks is using consensus algorithms that prioritize transaction processing based on their importance and urgency. An example can be the Proof-of-Importance (PoI) consensus algorithm which assigns higher priority to transactions that involve critical medical data such as emergency cases, while lower priority to less critical data[19][16].

Sharding can also enhance the scalability of blockchain in healthcare. It divides the blockchain network into smaller more manageable parts (shards) which is allowing for parallel processing of transactions and reducing the burden on the network.

Several studies have investigated the scalability of blockchain technology in healthcare. In a study by B. Fan et al., the authors proposed a hybrid consensus algorithm that combines Proof-of-Work (PoW) and Proof of X-repute (PoXR) to improve the scalability and security of blockchain in healthcare. The results showed that the hybrid consensus algorithm can reduce the block generation time and increase the transaction throughput[16].

Another study is by M. Shbier et al. who proposed a sharding-based approach to improve the scalability of blockchain in healthcare. The authors developed a prototype system that uses sharding to divide the network into smaller segments, allowing for faster transaction processing and reduced network latency[18].

Although scalability is a critical issue for the widespread adoption of blockchain technology, current researchs on scalability solutions, such as off-chain/on-chain techniques, consensus algorithms, and sharding etc. show optimistic results in improving the efficiency and throughput of blockchain systems.

1.3 Challenges

1.3.1 Scability Challanges

Because of blockchain systems are distributed, it requires consensus among a large amount of nodes which validate transactions. The main scability challenges that blockchain systems face to are block size limitations, high transaction volumes, high frequency trading and number of transactions per second, the number of nodes in the network, and the need of protocols.

One of the most significant challenges is the block size which refers to the maximum capacity for a block to be filled up with transactions. If the capacity is exceeded, the block will be rejected by the network[2]. This restriction can cause longer confirmation times and high transaction costs which will have a negative effect in usability of the blockchain network. For example, the Bitcoin network has a block size limit of 1 MB, which limits the transaction throughput to approximately 7 transactions per second (tps) [7].

Another scalability issue of the blockchain systems is that the high volume of transactions which blockchain networks should handle. As long as the blockchain networks grows and getting larger, the number of transactions which are being processed can increase exponentially, potentially exceeding the network's capacity. For example, the Ethereum network experienced high transaction volumes during the 2017 ICO boom, leading to congestion and higher transaction fees [6].

Additionally, the number of nodes in the network can also have an affect on the scalability of blockchain systems. With the increased number of the nodes, the complexity of achieving consensus among the nodes increases as well, which can lead to higher network latency and potentially lower transaction throughput. For example, the Bitcoin network has over 10,000 nodes worldwide, which can make achieving consensus more difficult[5].

Moreover, in blockchain networks, there is a need of protocols being used which has a natural impact on overall scalability of the network. Some consensus protocols like Proof-of-Work(PoW) that requires significantly high computational resource to validate transactions within the network which is limiting the transaction throughput of the network. On the other hand another protocol such as Proof-of-Stake(PoS) requires less computational power but it might come across to scalability issues when handling a large number of transactions[10].

As mentioned before, in larger blockchain systems, scalability is a big challenge. Block size limitations, high transaction volumes, the number of nodes, and the protocol used are some of the significant factors that can impact the scalability. However, there are several solutions that some of them will be discussed in the following section. These solutions are developed to address these challenges, which can pave the way for the widespread adoption of blockchain technology in the future[10].

1.3.2 Current challenges of blockchain and IoT in healthcare

The Internet of Things (IoT) is associated with the network of physical objects such as medical devices, Radio-Frequency, home appliances, and vehicle. And Internet of Medical thing aka. IoMT is the health customized version of IoT in which a doctor can measure various parameters of the patient's health such as heart rate, body temperature, and oxygen level remotely and immediately by means of different sensors placed in or on the patient's body[3]. Blockchain technology, which is a distributed ledger, can bring potential benefits to IoMT applications. It can solve major problems, for example the point of failure of the cloud in IoMT applications. And blockchain has the potential to provide security and protect against losses and risks and also helps preserve privacy[13][3].

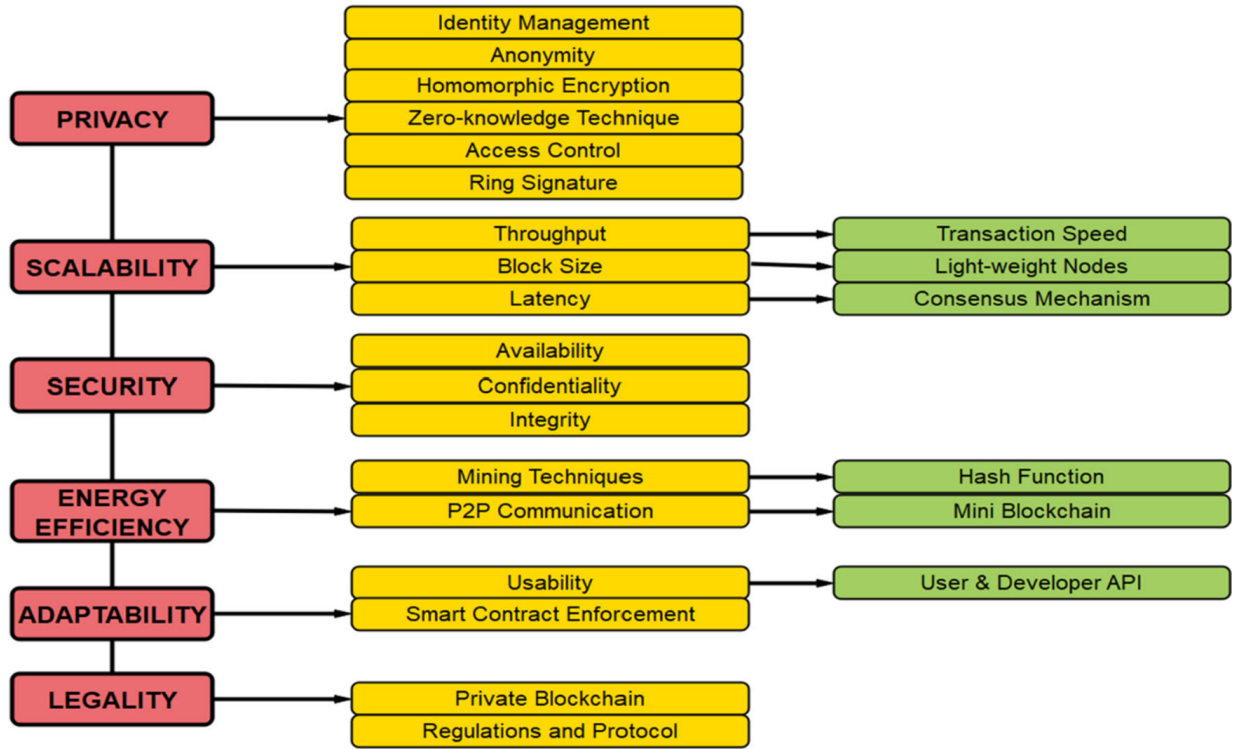


Figure 1.1: Current challenges of blockchain and IoT in healthcare

Blockchain and the Internet of Things have a great potential in the healthcare industry by enabling secure, decentralized data sharing and enhancing patient care[13]. However, the implementation of these technologies also presents several challenges that must be addressed to fully realize their potential, such as lack of standards and regulations, scalability, firmware and hardware vulnerabilities, lack of trusted authorities and data feeds, and challenges in smart contracts such as time-stamp dependence, dependence on the order of transactions, and call stack depth[3].

One significant challenge is the interoperability of healthcare systems. Healthcare data is often binned in different systems, making it difficult to share across different platforms. This interoperability challenge is further complicated when combining blockchain and IoT technologies, which have their own unique protocols and standards[29].

Another challenge is the scalability of blockchain and IoT systems. As discussed previously, blockchain systems face scalability issues related to block size limitations, high transaction volumes, the number of nodes, and the protocol used. Similar to this, IoT devices can generate vast amounts of data, which can quickly overwhelm the network's capacity and storage capabilities.

Data privacy and security are also significant challenges for blockchain and IoT in healthcare. While blockchain technology provides a high level of security through its decentralized and immutable nature, it is still vulnerable to attacks. To add that, IoT devices can be easily compromised, making them susceptible to security breaches and data leaks.

Regulatory compliance is another challenge. Healthcare is a heavily regulated industry,

and any new technology must comply with existing regulations and privacy laws. Compliance with regulations, such as HIPAA, can be difficult to achieve when implementing blockchain and IoT systems[21].

Finally, blockchain and IoT systems in healthcare face adoption challenges due to the resistance to change and lack of standardization. Healthcare professionals may be hesitant to adopt new technologies that could disrupt their workflow, and without industry-wide standards, the implementation of these technologies may vary significantly across different organizations and regions[21].

While MIIoT has immense possibility for the healthcare sector that their implementation also presents several challenges related to interoperability, scalability, data privacy and security, regulatory compliance, and adoption. Addressing these challenges will require collaboration between industry stakeholders, regulators, and technology developers to develop solutions that maximize the benefits of these technologies while minimizing their risks[21].

1.3.2.1 Block Size

Block size is one of the primary scalability challenges facing blockchain technology. The block size refers to the maximum amount of data that can be stored in a single block of transactions on the blockchain. As more transactions are processed and added to the blockchain, the block size grows bigger and it might lead to slower transaction processing times and higher costs.

The block size limit was originally introduced in Bitcoin which is the very first and therefore the most popular and well-known blockchain, to prevent spam attacks and ensure the stability of the network. However, as the popularity of Bitcoin and other blockchains grew, the block size limit became a bottleneck for transaction processing.

At first, it was designed with a block size limit of 1 Mb, therefore this technology became very famous for providing secure transactions without the need for third-party validations. However, like other systems, this one is also not perfect. Whenever a new block wants to join the chain, it must download the whole chain to learn about the chain and to be able to validate transactions. This not only requires vast memory but also has become time-consuming as well. Because it is based on verification of each and every transaction by mining nodes. So, large blockchains suffer from scalability issues and therefore there are some solutions implemented which can minimize mining latency and provide better scalability in blockchain[1] [7].

One example of the impact of block size limitations can be seen in the Bitcoin network. In 2017, the average transaction fee on the Bitcoin network spiked to over \$50 due to the high demand for transactions and limited block size. This led to long transaction processing times and increased frustration among users[5].

For this block size challenge, some blockchain networks have implemented solutions such as increasing the block size limit or implementing off-chain scaling solutions or using alternative consensus mechanisms that require less data storage. As an example, the

Challenges	Solutions
1. Block Size 2. High Volume of Data 3. Transactions 4. Number of Nodes	Storage Optimization
1. Protocol 2. Block Size	Redesigning Blockchain

Figure 1.2: Scalability challenges and solutions in healthcare blockchain system

Bitcoin Cash network increased its block size limit from 1 MB to 32 MB, allowing for more transactions to be processed in a single block.

However, increasing the block size limit can also lead to other scalability challenges, such as slower transaction verification times and increased centralization of the network. Increasing the block size limit can provide short-term relief, although it is essential to consider other solutions and balance the need for larger block sizes with the need for a secure and decentralized network.

1.3.2.2 Number of Nodes

Increase in the number of nodes is another significant scalability challenge facing blockchain technology. A node is a component on a blockchain-based system. It is the foundation of the technology which represents each entity connected to the network. Logarithmically, if more nodes are added to the network, the internode latency increases with every additional node. Additionally, the increasing number of nodes leads to an increase in computational resources in an IoT environment[2].

Nodes are individual computers or devices that participate in the validation and storage of transactions on the blockchain. The more nodes that are involved in a blockchain network, the more decentralized and secure the network becomes. However, as the number of nodes increases, the computational requirements for processing and verifying transactions also increase, leading to potential scalability challenges.

One example of the impact of the number of nodes on blockchain scalability can be seen in the Ethereum network. Ethereum is a blockchain network that supports the development of decentralized applications, and its popularity has led to a significant increase in the number of nodes participating in the network. As of April 2021, there were over 12,000 nodes on the Ethereum network, a significant increase from the approximately 5,000 nodes in 2019[6] [31].

While a larger number of nodes can increase the security and decentralization of the network, it can also lead to slower transaction processing times and increased costs. For

example, in 2021, the average transaction fee on the Ethereum network reached an all-time high of over \$70 due to the high demand for transactions and limited capacity of the network[6][31].

As a solution to this challenge, some blockchain networks has the implementation of sharding, which divides the network into smaller subsets or shards and with each shard, it is processing a subset of transactions. This can increase the capacity of the network and reduce the computational requirements for processing transactions and allow for more nodes to participate in the network while maintaining scalability. It has also some drawbacks such as increased complexity and potential security risks. Therefore, it is essential to carefully consider the trade-offs between scalability, security, and decentralization when addressing the challenge of the number of nodes in blockchain networks[30].

1.3.2.3 Transactions

Transaction is another challenge facing blockchain technology. As the number of transactions on a blockchain network increases, the time and resources required to process each transaction can also increase, leading to potential scalability challenges.

In the healthcare field, the connection of large communities connected is expected, which will involve the use of blockchain. Therefore, a high amount of transactions will be generated and recorded due to the increased usage of blockchain. The blockchain- distributed ledger will append new transactions after the agreement of multiple entities on every transaction linked to the transactions in the ledger. Although this process seems complex, it is effective as a limited size of the blockchain is given. Pre-authorization payments are even used by some parties using short-term data sharing to decrease the amount of memory usage in the blockchain[2].

One example of the impact of transaction scalability on blockchain networks can be seen in the Bitcoin network. Bitcoin is a decentralized digital currency that operates on a blockchain network, and its popularity has led to a significant increase in the number of transactions on the network. As of April 2021, the average daily transaction volume on the Bitcoin network was over 300,000 transactions, a significant increase from the approximately 300 transactions per day in 2010. As the transaction volume on the Bitcoin network has increased, so have the transaction processing times and fees. The Bitcoin network has a limited capacity to process transactions, with each block in the blockchain having a maximum size limit of 1 MB. As a result, the Bitcoin network can only process a limited number of transactions per second, leading to potential delays and increased transaction fees during periods of high demand[27] [28].

Some blockchains solved this issue with off-chain solutions such as payment channels or sidechains. These solutions involve moving some transactions off the main blockchain network, allowing for faster and more cost-effective processing of transactions. For example, the Lightning Network is an off-chain payment protocol built on top of the Bitcoin network that allows for near instantaneous and low-cost transactions. As before, it also has advantages and disadvantages. Some issues can be increased complexity and potential security risks[9].

1.4 Solutions

There are several solutions will be discussed regarding scalability issues in blockchain technology. Some of these solutions include:

- **Off-chain solutions:** These involve moving some transactions off the main blockchain network to allow for faster and more cost-effective processing of transactions.
- **Sharding:** This involves breaking the blockchain into smaller parts (shards) that can be processed in parallel by different nodes.
- **Consensus algorithms:** New consensus algorithms such as Proof of Stake (PoS) and Delegated Proof of Stake (DPoS) are being developed to improve the speed and scalability of blockchain networks.
- **Network optimization:** Improving the network infrastructure and optimizing network protocols can help improve the speed and scalability of blockchain networks.

Overall, there is no single solution to address scalability issues in blockchain technology, and it is likely that a combination of solutions will be needed to achieve significant improvements in scalability while maintaining security and decentralization.

1.4.1 Off-Chain Approach

Off-chain solutions involve moving some of the transaction processing off the main blockchain to improve performance. This approach can be beneficial in improving the scalability of blockchain networks without sacrificing security and decentralization.

The main advantage of off-chain solutions is that they can significantly reduce the burden on the blockchain network by moving some of the transaction processing off-chain. This can help to improve the performance of the network by reducing the load on individual nodes and reducing the transaction processing times. By moving some of the transaction processing off-chain, the blockchain can handle a larger number of transactions while maintaining its security and decentralization.

Lightning Network can a good example for this solution. The Lightning Network is a second-layer network built on top of the Bitcoin blockchain. It allows for near-instant and low-cost transactions by moving some of the transaction processing off-chain. The Lightning Network achieves this by creating a network of payment channels between users. These payment channels allow users to transact with each other without having to broadcast each transaction to the entire blockchain network. Instead, the transactions are only recorded on the blockchain when the payment channel is closed[9] .

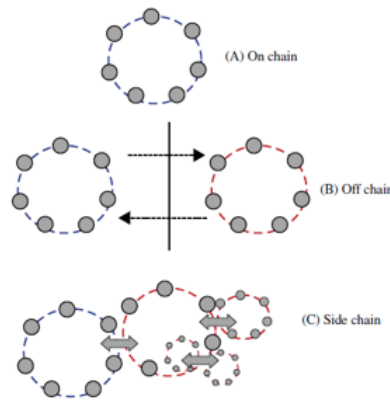


Figure 1.3: On-Chain, off-Chain, side-Chain

1.4.2 On-Chain Approach

On-chain scalability solutions refer to changes or improvements made directly to the blockchain protocol in order to increase the capacity of the chain for processing transactions.

Some of the of on-chain scalability solutions are the following:

- **Specific Consensus Mechanisms:** Different consensus mechanisms can be used to increase the capacity of a blockchain network.
- **Sharding:** it is a scalability solution that involves partitioning the blockchain into smaller parts called "shards." Each shard can process transactions independently, allowing for parallel processing and increased capacity. Ethereum is currently exploring sharding as a potential solution to its scalability issues.
- **DAG-Based Distributed Ledger Structures:** Directed Acyclic Graph (DAG) based distributed ledger structures, such as the one used by IOTA, do not use blocks to store transactions. Instead, transactions are linked together in a non-linear manner, allowing for more efficient processing and increased capacity.
- **Parallel Blockchain Extension:** Parallel blockchain extension involves the creation of multiple chains that operate in parallel, rather than a single chain.
- **Collective Leaders:** In some blockchain networks, multiple nodes may collaborate to act as a "collective leader," responsible for validating transactions and producing new blocks.

1.4.3 Storage Optimisation

Storage optimization is one of the key aspects of blockchain scalability. With increasing transaction volume and block size, blockchain storage requirements can become significant and can become a bottleneck for the overall performance of the system.

- Blockchain modelling: Blockchain modelling involves creating a new blockchain architecture that is specifically designed for high scalability.
- Read mechanism: One strategy for improving scalability is to optimize the read mechanism, which involves retrieving data from the blockchain.
- Write mechanism: Another strategy for improving scalability is to optimize the write mechanism, which involves adding new data to the blockchain. The Lightning Network, for example, uses off-chain channels to enable faster and more efficient transactions, reducing the number of write operations required on the main blockchain[9].
- Bi-directional network: The bi-directional network is a strategy for improving scalability by allowing for both horizontal and vertical scaling. A bi-directional network allows for the addition of new nodes to increase the number of transactions that can be processed horizontally, while also allowing for the optimization of individual nodes to increase processing power vertically.

Strategies such as blockchain modelling, read and write mechanisms, and bi-directional networks offer different approaches to redesigning the blockchain to increase scalability. However, implementing these strategies can be challenging, as they require changes to the underlying blockchain architecture and protocols.

1.5 Conclusion

To conclude, scalability is a significant challenge in blockchain technology, and it has the potential to limit the adoption of this innovation of this technology in various fields and one of them is healthcare.

Constantly, the number of nodes and transactions on a blockchain network increases which causes the growth of the size of the blockchain. Eventually it is leading to longer transaction times and higher processing costs, which can significantly affect the usability of blockchain technology.

Various solutions have been proposed to overcome the scalability issue. Some of them are on-chain and off-chain solutions, redesigning blockchain and storage optimisation techniques. These solutions offer promising ways to improve the performance and scalability of blockchain networks, however they also have their own challenges such as increased complexity, reduced security, and potential for centralization.

In order to find a sustainable and secure solution to the scalability issue in blockchain systems, there is still need to further research and development. The use of advanced technological solutions, collaboration between industry players, researchers, and regulators are crucial to finding a scalable and secure solution.

In spite of the challenges, this new and innovative technology still remains as an optimistic innovation with high potential in designing applications in various industries and sectors.

Systems can profit from blockchain, for example with improved security, transparency, and efficiency. Addressing the scalability issue will help unlock the full potential of blockchain technology and enable its widespread adoption.

Bibliography

- [1] Kaur, Gagandeep, Gandhi, Charu. (2020). Scalability in Blockchain: Challenges and Solutions. 10.1016/b978-0-12-819816-2.00015-0.
- [2] Mazlan, Ahmad, Mohd Daud, Salwani, mohd sam, Suriani, Abas, Hafiza, Rasid, Siti, Yusof, Fathi. (2020). Scalability Challenges in Healthcare Blockchain System - A Systematic Review. IEEE Access. 8. 1-1. 10.1109/ACCESS.2020.2969230.
- [3] Adavoudi Jolfaei, Amirhosein, Aghili, Farhad, Singelee, Dave. (2021). A Survey on Blockchain-Based IoMT Systems: Towards Scalability. IEEE Access. PP. 1-1. 10.1109/ACCESS.2021.3117662.
- [4] Ghosh, P.K.; Chakraborty, A.; Hasan, M.; Rashid, K.; Siddique, A.H. Blockchain Application in Healthcare Systems: A Review. Systems 2023, 11, 38. <https://doi.org/10.3390/systems11010038>
- [5] Bitcoin.org, Bitnodes, 2022. <https://bitnodes.io/>.
- [6] Ethereum Foundation, Ethereum Scaling Ethereum to Billions of Users, 2017. <https://www.coinbase.com/blog/scaling-ethereum-crypto-for-a-billion-users>.
- [7] Nakamoto, S. Bitcoin: A Peer-to-Peer Electronic Cash System, 2008. <https://bitcoin.org/bitcoin.pdf>.
- [8] R. Ali, S. U. Khan, S. A. Madani, and M. Imran: Blockchain-based e-Health System: A Comprehensive Review; technical report, 2018, 41025-41044
- [9] Poon, J., Dryja, T. (2016). The Bitcoin Lightning Network: Scalable Off-Chain Instant Payments. White Paper.
- [10] A. Dorri, S. S. Kanhere, R. Jurdak, and P. Gauravaram:Blockchain for IoT Security and Privacy: The Case Study of a Smart Home; IEEE Communications vol. 56, no. 9, pp. 149-155, 2018.
- [11] C. Natoli, V. Gramoli, A. Anceaume, and G. F. D. L. M. Mosseri, "The Blockchain Anomaly," in Proceedings of the ACM Symposium on Principles of Distributed Computing, 2016, pp. 273-282.
- [12] P. Kshetri, "Blockchain's Roles in Meeting Key Supply Chain Management Objectives," International Journal of Information Management, vol. 39, pp. 80-89, 2018.

- [13] Meier, Pascal, Beinke, Jan, Fitte, Christian, Schulte To Brinke, Jan, Teuteberg, Frank. (2021). Generating design knowledge for blockchain-based access control to personal health records. *Information Systems and e-Business Management*. 19. 10.1007/s10257-020-00476-2.
- [14] R. Ivan, A. Khadivi, and M. J. Roberts, "Blockchain in Healthcare: A Systematic Review," *Healthcare*, vol. 7, no. 2, p. 56, 2019.
- [15] P. M. Singh, R. Singh, and A. K. Singh, "Blockchain Technology in Healthcare: A Comprehensive Review and Directions for Future Research," *International Journal of Healthcare Information Systems and Informatics*, vol. 15, no. 3, pp. 20-38, 2020.
- [16] B. Fan, J. Li, H. Li, and Y. Li, "A Hybrid Consensus Algorithm for Scalable Blockchain-Based Healthcare System," *IEEE Access*, vol. 8, pp. 136997-137005, 2020.
- [17] Yujin Han, Yawei Zhang and Sten H. Vermund, "Blockchain Technology for Electronic Health Records," 2022, doi: 10.3390/ijerph192315577.
- [18] M. Shbier, T. Taleb, and H. Elgala, "A Scalable Blockchain-Based Architecture for Healthcare," in *Proceedings of the 6th IEEE International Conference on Future Internet of Things and Cloud*, 2018, pp. 116-122.
- [19] Y. Yang, H. Wang, and K. Yang, "A Consensus Algorithm for Blockchain-Based Healthcare Systems," in *Proceedings of the 3rd International Conference on Computer Science and Application Engineering*, 2020, pp. 1-7.
- [20] Schmeelk S, Kanabar M, Peterson K, Pathak J. Electronic health records and blockchain interoperability requirements: a scoping review. *JAMIA Open*. 2022 Jul 27;5(3):ooac068. doi: 10.1093/jamiaopen/ooac068. PMID: 35911668; PMCID: PMC9329659.
- [21] A Review on Adopting Blockchain and IoT Technologies for Fostering the Circular Economy in the Electrical and Electronic Equipment Value Chain, 2023.
- [22] Linn, L. A., Koo, M. B. (2019). Blockchain for health data and its potential use in health IT and health care related research. *Clinical and Translational Science*, 12(2), 98-105.
- [23] Sharma, N., Chen, X. (2021). Blockchain for healthcare: Opportunities and challenges. *Journal of Medical Systems*, 45(4), 1-10.
- [24] P. D. Grover, A. Kumar, and S. Gupta, "Scalability challenges and solutions in blockchain-based healthcare," *Journal of Medical Systems*, vol. 43, no. 10, pp. 1-10, Sep. 2019.
- [25] Xi, P.; Zhang, X.; Wang, L.; Liu, W.; Peng, S. A Review of Blockchain-Based Secure Sharing of Healthcare Data. *Appl. Sci*. 2022, 12, 7912. <https://doi.org/10.3390/app12157912>

- [26] Pilkington, Marc, Blockchain Technology: Principles and Applications (September 18, 2015). Research Handbook on Digital Transformations, edited by F. Xavier Olleros and Majlinda Zhegu. Edward Elgar, 2016, Available at SSRN: <https://ssrn.com/abstract=2662660>
- [27] Antonopoulos, A. M. (2014). Mastering Bitcoin: Unlocking Digital Cryptocurrencies. O'Reilly Media, Inc.
- [28] Bonneau, J. (2015). Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction. Princeton University Press.
- [29] Reegu, Faheem, Daud, Salwani, Alam, Shadab. (2021). Interoperability Challenges in Healthcare Blockchain System -A Systematic Review. Annals of the Romanian Society for Cell Biology. 25. 15487-15499.
- [30] Buterin, V. (2018). The Meaning of Decentralization. Vitalik Buterin's website.
- [31] Wood, G. (2014). Ethereum: A Secure Decentralised Generalised Transaction Ledger. Ethereum Project Yellow Paper, 151.

1.6 Appendix I

Blockchain scalability solution algorithms with explanations

Method / Model	Summary
Reference pointer of FHIRChain [12]	Enhancing the size of the reference pointer of the data to make it scalably smaller.
Mini blockchain [36]	Enhance the scalability by removing the requirement for storing the entire blockchain.
VerSum [37]	Use smaller computational resources by enabling the outsourcing computations over a high number of public logs to a collection of servers.
FHIRChain [12]	It was designed for secure and scalable sharing of clinical data. FHIRChain produces lightweight reference pointers without sharing actual databases and as a solution for network coverage, especially in the clinics or telemedicine clinics in rural areas.
HealthChain [22]	Its implementation of smart contracts ensures proper authorization and specifies the privileges on its permitted network.
DeepLinQ [34]	Multi-layer blockchain architecture which improves flexibility, accountability, and scalability through on-demand queries, proxy appointment, subgroup signatures, granular access control, and smart contracts.
OmniPHR [38]	Latency remained stable despite the increase in the number of nodes, routing overlays, and backbone routers. Plus, significant number of messages was transmitted at the same time in the network.
Short-term data sharing [20]	Able to store and exchange encrypted metadata through a reference pointer and expiration configuration. Essential for healthcare services subscription or even in emergency situations.
Caching system [27]	This design improves throughput performance with the use of less FPGA power and work area. Besides, remarkable reduction of workload by 103 times was observable from the result.
Smart contract [5]	It has the potential of reducing fraud as a legal digital agent to create, store, and enforce contract. It builds on trust between publisher and subscriber.
Tokenization [12]	This method will complement the process of encryption by generating a pair of keys for user and registers.
Chord algorithm [28]	The algorithm provide service for OmniPHR distribution on the P2P network for adaptation to changes of workload, linkage for distributed network, and managing amount of workload.
Sharding [39]	It divides the mining network into smaller parallel chains. Therefore, each shard process is in parallel with other shards, leading to a micro block from each shard. Zilliga claimed that this process is linearly scalable to the expansion of the blockchain network.
Practical Byzantine Fault Tolerant (PBFT) consensus protocol [39]	Improved the finality of transactions and no confirmation is required for additional (next) block. However, only happened if the group consensus occurs in a sequential.
TrustChain [40]	Self-reinforcing trust are enabled by the preferring agents who have been helpful to the other agents in the past.
Lightning network [41]	The design of network of micropayment channels enables bitcoin scalability and micropayments to the near-instant transactions. It represents the transactions and funds transfer could be performed without the risk of counterparty theft.

Figure 1.5: Scalability solution algorithms in blockchain systems

Chapter 2

Promises of the 5G Network

Sabyasachi Banik

The introduction of fifth-generation (5G) mobile networks is anticipated to significantly alter how we use technology. This study looks at the 5G network's potential to offer hitherto unheard-of levels of connection, speed, and dependability. The report examines how 5G technology might revolutionize industries including healthcare, autonomous driving, augmented and virtual reality, and the Internet of Things. The study also analyzes the possible advantages of 5G for companies, such as improved productivity and efficiency. In addition, the difficulties and challenges of 5G are discussed, including worries about security and privacy. In the end, this article emphasizes how 5G has the ability to completely change how we communicate, work, and live. Yet, it also recognizes the need for continued research and development to guarantee that the technology is implemented in a secure, ethical, and fair manner.

Contents

2.1	Introduction	25
2.2	Background	25
2.2.1	Wireless Telecommunication System	25
2.2.2	5G? Why 5G?	26
2.2.3	History and Comparisons of 5G with different generations . . .	27
2.2.4	Contribution and Usage Prediction of 5G Globally	28
2.3	Facilitators	29
2.3.1	Different Enabling Scenarios	29
2.3.2	Timing Devices of 5G: MEMS timing device slowly replacing Quartz based counterparts	34
2.4	Critical Discourse	35
2.4.1	Visions for the Future	35
2.4.2	Challenges	38
2.4.3	5G, will you be able to keep your promises?	39
2.5	Discussion	40
2.5.1	Policy Principles for Global Policymakers	40
2.5.2	Open Issues, Future Directions and Motivation for Research . .	41
2.5.3	6G and Beyond!	42
2.5.4	Intriguing 5G Questionnaire Set	43
2.6	Conclusion	45

2.1 Introduction

The most recent and cutting-edge cellular network technology, known as the fifth generation (5G), has the potential to completely transform the way we work, play, and live. With its breakneck data rates, incredibly low latency, and widespread connection, 5G is anticipated to open up a world of new opportunities for organizations, governments, and people alike. The business world has been resonating with the potential of 5G for years. In comparison to 4G or Wi-Fi 6, 5G offers new possibilities by laying the groundwork for widespread ultra-fast internet [5]. The full potential of technical advancements including as artificial intelligence (AI), edge computing, the Internet of Things (IoT) and extended reality (XR) will be realized when 5G is combined with them. This, again, if you believe the hype, will lead to a whole new Internet of Things, where everything from toasters to dog collars to dialysis pumps to running shoes will be connected [15]. Powered by advanced technologies, algorithms, and devices, 5G is becoming the foundation for the digitalization of industry and society [3].

This report typically aims to examine the extent to which the promises of 5G have been realized in both technical and application dimensions, to discuss difficulties and challenges, and to provide an outlook on developing next-generation communication technologies. It starts by claiming how 5G outperforms all the other previous generations and its usage prediction in different sectors by the end of the decade. This report also looks into different enabling scenarios of 5G including its development, visions, and compare the reality with the promise. Finally, future challenges and potential issues facing the new generation of communication technologies will be discussed to bring insights for future directions, open issues and further research and development.

2.2 Background

The following section is intended to provide an overview of wireless telecommunication systems- mobile communication systems in particular. 5G properties and requirements along with its difference from OSI protocol stack have been discussed. Lastly, this section gives a comparison of 5G with other previous generations and the usage prediction of 5G Globally.

2.2.1 Wireless Telecommunication System

A wireless telecommunication system is a communication network that uses wireless signals instead of physical cables or wires to send and receive information between devices. Today's wireless and telecommunication technologies work hand in hand to provide the general public access to wireless communication that makes it simple for people to communicate no matter where they are in the globe. Information is sent over the air utilizing electromagnetic waves such as satellite, RF (Radio Frequency), and infrared (IR). For instance, Bluetooth, Wi-Fi, GPS, satellite television, wireless computer components, and

wireless phones with 3G and 4G networks [15]. Wireless technology offers speed, flexibility, and network efficiency. It has become a powerful tool for tech-savvy generation as it facilitates easy information sharing and boosts productivity. One gets the freedom to roam around freely without worrying about the internet connection and still stay connected.

Mobile Communication Systems which is a sub-section of Wireless Telecommunication Systems truly started to take a flyer with the introduction of GSM(Global System for Mobile Communications). Based on TDMA(Time Division Multiple Access) it is basically referred as 2G. Then sequentially, we had UMTS(Universal Mobile Telecommunication System), HSDPA(High Speed Downlink Packet Access) and LTE(Long Term Evolution). UMTS refers to a group of radio technologies associated with third Generation of Cellular Network(3G). HSDPA is an enhanced 3G mobile communications protocol also known as 3.5G. After LTE(sometimes referred as 4G LTE) came into being. With the increasing demand for increased capacity and improved network reliability, the world is slowly switching towards the fifth Generation of Cellular Network (5G) which has many promising visions and benefits to look upon. Towards the end, this report would also peek into 6G, considering what more it has to offer and how it can outperform 5G.

2.2.2 5G? Why 5G?

Next-generation mobile communication technology designed to provide ultra low latency and higher data speeds than the previous generation Long Term Evolution (LTE).

Key Requirements[6]: 1. The ability to sustain huge connections and capacity 2. Support for a growing range of users, services, and apps, all of whom have wildly varying needs for both work and life, 3. Use of all non-contiguous spectrum in a flexible and effective manner for various network implementation situations

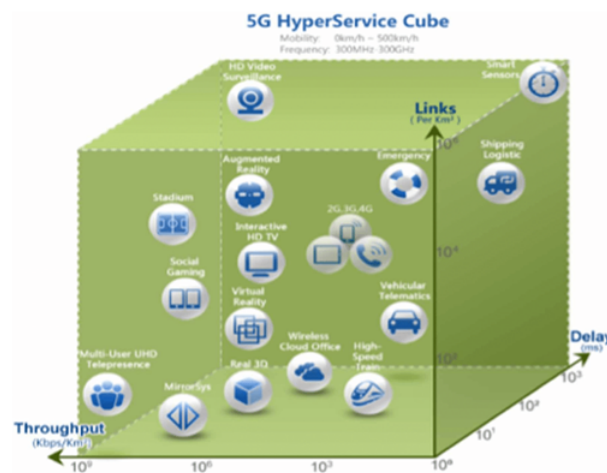


Figure 2.1: 5G HyperService Cube [6]

Difference between OSI vs 5G protocol stack:

Table 2.1: OSI stack vs 5G protocol stack

APPLICATION LAYER	APPLICATIONS of SERVICE
PRESENTATION LAYER	
SESSION LAYER	OPEN TRANSPORT PROTOCOL
TRANSPORT LAYER	UPPER NETWORK LAYER
NETWORK LAYER	LOWER NETWORK LAYER
DATA LINK LAYER	
PHYSICAL LAYER	OPEN WIRELESS ARCHITECTURE
OSI Stack	5G Network Stack

Relation to OSI Model defined [16]:

Open Wireless Architecture Layer(OWA): OWA layer, also known as Open Wireless Architecture layer, is the abbreviation for this term. It serves as the OSI stack's physical layer and data connection layer. **Network Layer:** Data is routed via the network layer from the IP device at the source to the IP device or system at the destination. There are lower network layers and higher network layers. **Transport Layer:** Transport Layer Open It combines the session layer and transport layer features. **Application Layer:** It annotates the data with the appropriate format as needed. The data is also encrypted and decrypted by it. It chooses the optimal wireless connection for a particular service.

2.2.3 History and Comparisons of 5G with different generations

Table 2.2: Comparisons of different generations

0G	1G	2G	3G	4G	5G
-Calling Zones -No handover	-Analog signalling -Individual subscribers possible -Handover between cells	-Higher capacity -Fully digital transmission and signalling	-Integration of voice and video in cellular networks -World-wide compatibility for terminals and base stations	-High speed fully digital -Backward compatibility to 3G	-Ultra short response times -Up to 100 times faster than 4G) -Parallel Streaming possible

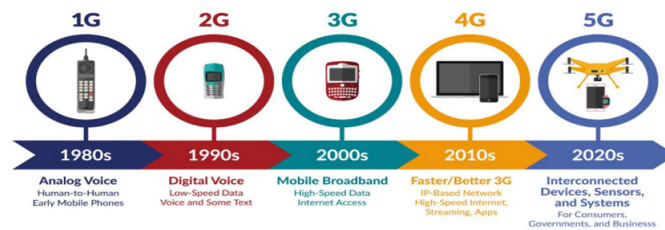


Figure 2.2: Importance and growth of 5G Network [11]

We also had 2.5G, 3.5G(HSDPA) in between and one important point to be considered is that these half generations did not come in sequence. They showed up whenever new requirements were felt necessary keeping in mind the existing financial conditions.

2.2.4 Contribution and Usage Prediction of 5G Globally

With the expected number of increased connections over the years in aggregate adoption of 5G is expected to add 1.3 trillion USD to the Global GDP. By 2025, 20% of connections will be 5G connections, with developed Asia, North America, and Europe seeing the highest adoption rates. By 2025, there will be 1.2 billion additional individuals accessing mobile internet. Operators anticipate investing \$1.1 trillion in mobile Capex globally between 2020 and 2025, with around 80% of that amount going toward 5G networks, to enable this generational change and further increase consumer engagement.

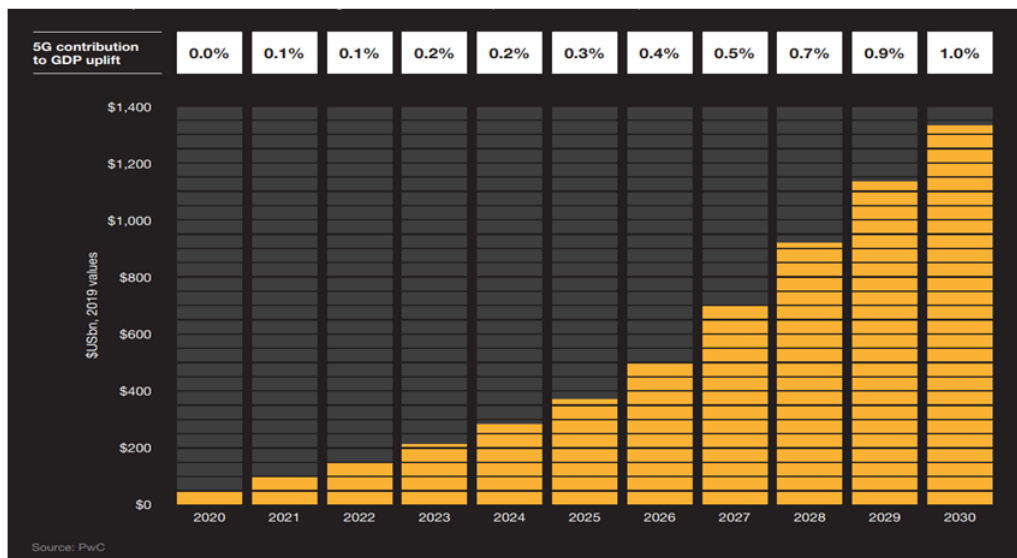


Figure 2.3: Projected contribution of 5G to global GDP,2020–'30 (US\$bn,2019 values) [5]

It is predicted that The 5G generation will be the first in mobile history to have a greater effect on businesses than on consumers [2]. According to one prediction, 5G would create twenty-two million new employment in the United States alone and inject twelve trillion dollars into the global economy by 2035. We are informed that the 5G era will herald the fourth industrial revolution [9]. Overall, it is expected that 5G will have a transformative impact on many industries and sectors including healthcare, manufacturing, transportation and more which will be looked in detail in section 2.4.1. By 2030, we can expect to see a wide range of innovative-5G-enabled applications and use cases that are yet to be fully realized. At the geographical level, North America will have the largest percentage increase in GDP from 5G, followed by Asia and Oceania, Europe, the Middle East, and Africa (EMEA). It is interesting to note that for most of European Countries either 5G is still not on the radar or they are not yet convinced by the promises and visions of 5G.

McKinsey and Co. predicts that by 2030 only one-quarter of the world's population will have access to high-band 5G coverage. There possibly could be two reasons behind this



Figure 2.4: Projected economic impact by country by 2030 (US\$, 2019 values) [5]

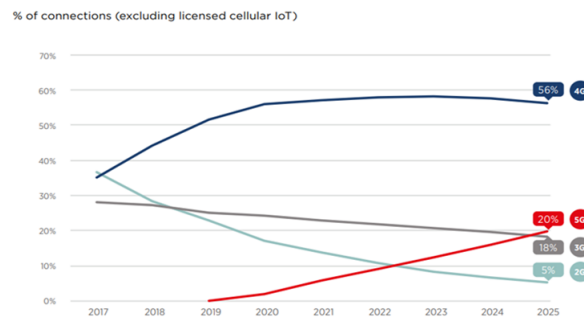


Figure 2.5: Percentage of connections(excluding licensed cellular IoT) [2]

prediction. Firstly, when we talk about world's population, we also consider many regions of under-developed and developing nations which are yet out the reach of proper electricity, let alone stable internet connection. Secondly, 5G still has to overcome quite some challenges to be able to deliver its promises and realize its vision. These challenges will be looked upon in detail in section 2.4.2. As it can be seen in figure 2.5, Half of all connections are already on 4G, and 5G has started to make a difference since 2020, but is still only expected to reach 20% by 2025.

2.3 Facilitators

This section provides an overview of different 5G enabling scenarios amongst which a bit more emphasis has been laid on Spectrum Sharing and Flexibility. Secondly Timing devices of 5G has been looked at in detail.

2.3.1 Different Enabling Scenarios

The introduction of 5G technologies has considerably helped society and prompted a plethora of academic research. The smoother implementation of 5G in the market can be supported by different enablers shown in 2.6

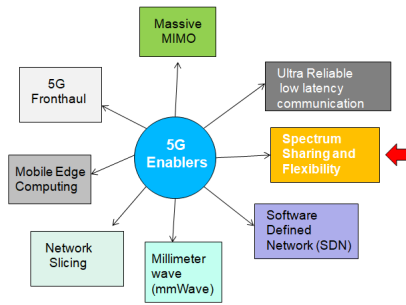


Figure 2.6: 5G Enablers

2.3.1.1 Massive MIMO (Multiple Input Multiple Output):

The concept underlying Massive MIMO is the use of a large number of antennas to allow great diversity and compensate for route loss for high-frequency transmission. This technique is also referred to as advanced antenna, very big MIMO, Hyper MIMO, full dimension MIMO, and large-scale antenna systems. They include employing a huge number of antennas per site to serve an area that is ten times bigger in one cell, increasing the number of customers serviced concurrently. Massive MIMO base stations can interact with many users at once over the same frequency, resulting in great scale efficiency, far greener technology, and much-increased edge-user performance. Massive MIMO offers low latency and enables EE communication, making it appropriate for the development of 5G [17].

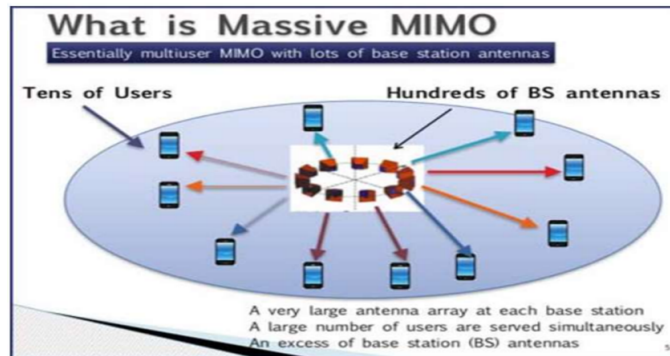


Figure 2.7: Massive MIMO; Source [IEEE]

2.3.1.2 Ultra Reliable low latency communication:

The ultra-low latency enabler, which can shorten processing times and transmission intervals and increase the bandwidth of radio resource blocks in which a certain amount of data is delivered, is another significant 5G promise [3]. Although the direct communication link (also known as D2D) provides low-latency transmission for devices nearby, this feature may also avoid radio transmitter queuing delays. The physical channel structure must be built for quick decoding at the recipient in order to realize this goal, and the MAC

must support instant access. By offering dimensional instant-access resource allocations, collision risks must be reduced. To meet this low latency criterion, several recent studies conducted in-depth evaluations.

2.3.1.3 Spectrum Sharing and Flexibility:

Given the value, cost, and high demand of the radio spectrum, effective management strategies are needed to maximize utilization [18]. Spectrum allocation allots a certain area of the spectrum to authorized users, or PUs, for a given purpose and duration. When PUs are not using the spectrum they have been given, it is unoccupied and available to other users, or SUs, who are in need. As a result, the owners of the limited spectrum resources squander them and do not fully use them. By allowing SUs to use the free spectrum on-demand, dynamic spectrum access has been offered as a solution to the issue of under-utilized spectrum [19]. It allows sharing the spectrum among users using spectrum sensing techniques and maximizing its utilization [20].

Cognitive radio (CR) is one efficient way of Spectrum Sharing. CR is a kind of wireless communication where a transceiver can discern between channels that are being used and those that are not. It's a new technology that can meet the 5G networks' rigorous spectrum requirements. Spectrum sensing, spectrum allocation, and spectrum access are the three key activities that makeup CR's functionality. SUs employ spectrum sensing to find unclaimed frequency bands that they may use without interfering with the owner's communications. Energy, matching filters, Euclidean distance, and cyclostationary-based detection are only a few of the spectrum sensing methods that have been suggested [20]. Then, after detecting the free spectrum holes, SUs compete to use that available spectrum based on the quality of service through distribution techniques [21]. These methods work by delivering each component of the available spectrum to a different user after being divided into smaller portions. They enable precise and effective distribution of the available spectrum among several SUs. SUs may access the identified spectrum for their transmission thanks to dynamic spectrum access. Afterwards, while keeping an eye out for any potential exit strategies, SUs transmit using the available frequency spectrum. Spectrum handoff or spectrum handover refers to the process of moving from one frequency band to another [22]. Spectrum handoff could be passive, active, or hybrid. To prevent interference during the passive handoff, SUs must instantly depart the frequency range and halt communicating when the PUs arrive back. Limited Spectrum Access is one of the top concerns for Operators(Figure 2.8) Requires Spectrum Policies and Regulation rules (will be discussed in Section 2.5.1)

2.3.1.4 Software Defined Network(SDN):

SDN has two defining characteristics: first, it separates the control plane (which decides how to handle the traffic) from the data plane (which forwards traffic according to decisions that the control plane makes). Second, it consolidates the control plane, so that a single software controls multiple data plane elements- Feamster et al. (2014): The Road to SDN. SDN remarkably provides Network Optimization. One or more controllers make up the

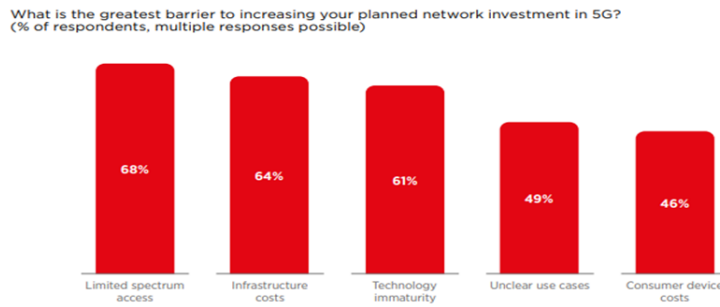


Figure 2.8: Spectrum is the top concern for operators in the 5G era [2]

control plane, which is thought of as the SDN network’s brain and center of integrated entire intelligence. This means we have a centralized system at hand. Yet, centralization poses issues with security, scalability, and flexibility on its own.

2.3.1.5 Millimetre Wave(mmWave):

The frequency increases as the wavelength becomes smaller. High-frequency waves are better at absorbing energy and resisting distortions. mmWave is one of the methods investigated in several studies to address the problem of spectrum shortage to realize 5G architecture. The mmWave frequency band, which spans from 30 to 300 GHz, is an ultra-high frequency band [23]. It delivers new services and the economic sector that may profit from 5G with exceptionally high data rates, ultra-high capacity, very big bandwidth, and very low latency. Furthermore, significant propagation loss, susceptibility to blocking, and directivity are characteristics of mmWaves systems. Even if these traits have great promise, these systems nevertheless face a number of difficulties, such as interference control, spatial reuse, dynamic control, integrated circuits, system design, high propagation loss, and anti-blockage [4].

2.3.1.6 Network Slicing:

A shared network domain, or collection of shared network and computing resources, is layered with several virtual networks via the process of network slicing [24]. It enables Operators to divide their physical network into many virtual, logical, and end-to-end networks called “slices” and better manage their resources. Slices are logical networks that span many physical networks. With independent devices, devoted resources, access, transport, and core networks, each network slice is isolated from the others. Network slicing also contributes significantly to economic profit by raising revenue and lowering CAPEX/OPEX costs for the network.

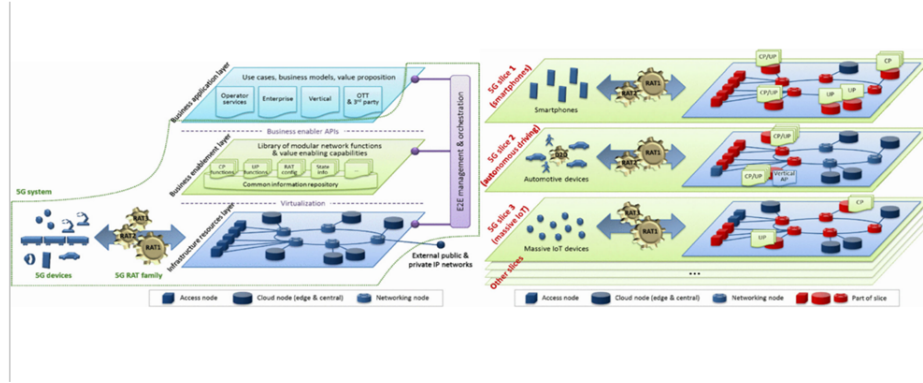


Figure 2.9: Multiple network slices over the same infrastructure [6]

2.3.1.7 Mobile Edge Computing:

The main idea is processing is closest to where data is captured. It allows moving cloud computing assets and network operations to edge nodes, such as base stations, gateways, access points, and user devices, which are located closer to mobile users. Low latency, energy-saving, context-aware computing, and enhanced security are all claimed benefits. Moreover, it offers security and resiliency. Context Delivery Network(CDN) is a typical use case of Mobile Edge Computing. CDN are geographically located groups of servers across the world aimed at caching content close to where each user is accessing the internet. Benefits of using CDN include improved load time, reduced bandwidth cost, increased content availability and redundancy and improved website security(mitigate DDoS attacks)

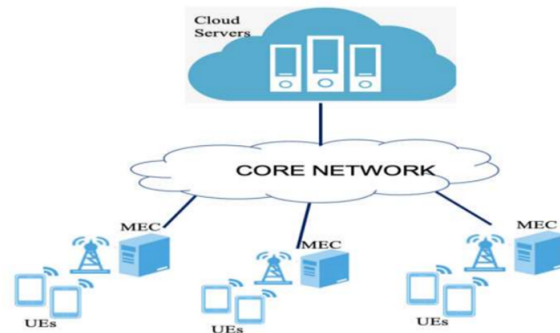


Figure 2.10: Mobile Edge Computing scenario [6]

2.3.1.8 5G Fronthaul:

Fronthaul portion of a C-RAN architecture comprises of links between the centralized baseband radio controllers and remote radio heads at the “edge” of cellular network. In recent years Fronthaul is becoming more essential as 5G is becoming more mainstream. One of the newest applications for passive optical networks(PON) is 5G fronthaul. PON

is an “Unpowered” Fiber Optic Network where the idea is to combine all signals onto a single fiber using WDM(Wavelength Division Multiplexing) and transmitted to end users via passive optical splitters. Using PON networks to complete the Fronthaul connections may decrease fiber count and enhance efficiency without sacrificing performance because of the bandwidth and latency requirements set by 5G.

2.3.2 Timing Devices of 5G: MEMS timing device slowly replacing Quartz based counterparts

All electronic systems, including communication networks, machinery, automobiles, and innumerable consumer electronics goods, rely on timing devices to function [8]. A timing chip may be compared to a pianist’s metronome in that it provides the artist with an accurate, consistent beat for a sharper, more precise musical performance. Despite the fact that timing technology is pervasive in our lives, relatively few individuals, with the exception of system engineers and architects, are aware of the crucial role clocks and oscillators have played in communications improvements over the last several decades. As we go towards the 5G era, timing technology is more crucial than ever. It is commonly known that 5G infrastructure requires the use of improved RF ICs, antenna arrays, amplifiers, beamforming, and beam management techniques. To be successful, 5G applications must choose the right temporal response.

The quartz crystals that silently resonate within our analog wristwatches serve as the basis for the majority of timing devices used in modern electronics. Resonators and oscillators made of quartz, a timing technology with a 70-year history, have served us well for decades in a variety of fields and applications [8]. However, the timing sector is experiencing a silent change. Quartz has been replaced by new versions of timing devices based on tiny, inventive MEMS resonators in applications that demand the greatest dependability and resistance to environmental stressors. When MEMS technology and analog circuits are merged, they offer a full-timing solution that is both more durable and smaller and consumes less power than comparable quartz-based devices.

Since they were originally made available for usage in the marketplace about 15 years ago, MEMS timing devices have steadily supplanted their quartz-based counterparts in a variety of demanding communications and networking applications, namely 4G LTE and 5G wireless networks. MEMS-based resonators, oscillators, and whole “clock-system-on-a-chip” devices operate orders of magnitude faster, more reliably, and longer than quartz-based equivalents. Because of these aspects, MEMS is the ideal precision timing technology for outdoor 5G macro- and small-cell base stations, allowing device makers and mobile carriers to fully realize the potential of 5G.

Customers’ happiness with 5G will be a factor in its success, along with its ability to deliver unmatched cellular speed and rock-solid dependability. There just isn’t a solution for lost calls or network failures brought on by soaring heat, high vibration, or abrupt shocks with 5G. Operators and customers must depend on 5G as a foolproof technology for applications like remote surgery and self-driving cars. Network infrastructure is already using MEMS-based timing technologies according to 5G equipment manufacturers.

MEMS timing devices are really used in more than ten distinct 5G applications now. It's about time: The 5G revolution is already well begun, and the most recent advancements in MEMS timing technology may contribute to the upcoming innovation and transformation of wireless networks.

2.4 Critical Discourse

In this section, the visions and goals of 5G have been discussed and an estimated GDP upliftment for different sectors from 5G has been shown. Later different challenges for 5G implementation have been looked upon.

2.4.1 Visions for the Future

5G aims to support increasing capacity demands with faster average download speeds. With 5G, in heavily crowded places or while far from Wi-Fi hotspots, it would be simpler to stream high-definition media. Other potential benefits include a) integrated management of the vast amount of connected devices in a smart society b) lower cost c) lower battery consumption and e) improved support of device-to-device communication. As can be seen in Figure 2.11 majority of respondents prioritized improved mobile data speed as the most important vision of 5G.

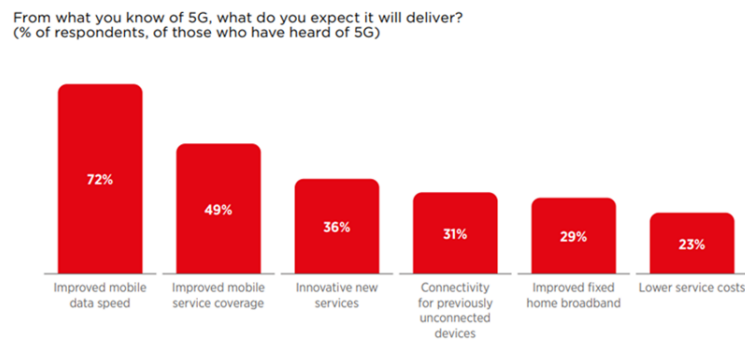


Figure 2.11: Respondents' priorities as visions for 5G [2]

2.4.1.1 5G in Health Care:

The economic benefits of 5G will be mostly attributed to the healthcare industry. By 2030, 5G-powered healthcare applications would boost the global Economy by US\$530 billion. The following points would be crucial for 5G in Health Care:

a) Fewer and shorter hospital stays: One area where 5G might allow cost savings and improved health outcomes is remote care. For instance, using 5G-enabled patient beds and medical equipment monitoring, as well as integrating artificial intelligence (AI) with 5G

sensor data to automate tasks and speed up patient handovers, would increase efficiency and productivity in hospitals.

b) Streamlined doctor-patient interactions: 5G-powered mobile health platforms will make it possible for health information to be collected and shared more quickly and accurately across healthcare systems as well as between patients and healthcare professionals. Also, it could make it possible for surgeons to treat patients using remote-controlled surgical robots or scanners that run on 5G. These tools will help physicians make better use of their time [5].

c) Drone-transported medical equipment and therapies: A 5G-capable drone may carry a defibrillator to the patient's location up to four times faster than an ambulance could. Moreover, drones are capable of picking up and transporting blood or tissue samples to a central laboratory for analysis as well as delivering drugs, blood, and immunizations quickly and across long distances. As can be seen in Figure 2.12, USA is expected to have the biggest upliftment to GDP from 5G in the Health Care sector followed by China, Japan and Germany.

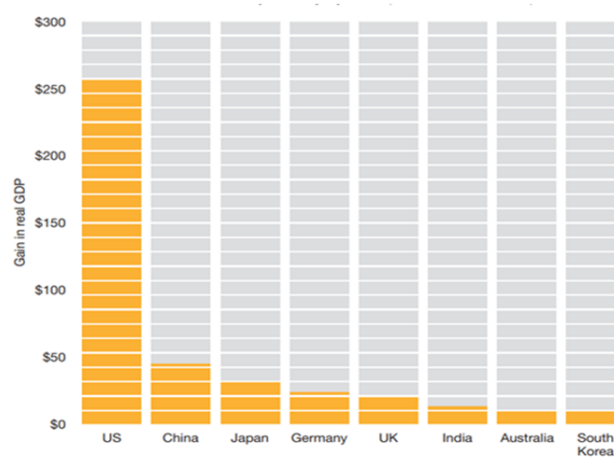


Figure 2.12: Country-wise GDP gain(US\$bn, 2019 values) from 5G in Health Care sector by 2030 [5]

2.4.1.2 5G in Smart Utilities:

Electricity and gas companies would be able to offer clients with a more in-depth picture of their energy consumption if "smart meters" and "smart grids" were improved. By 2030, 5G-powered smart utility management will increase global GDP by US\$330 billion. Better waste management and reduced solid waste, reduced water leakage are other areas where 5G will enable better use of its application. China and USA are expected to get most 5G boost to smart utility management by country by 2030 in terms of GDP.

2.4.1.3 5G in Industrial manufacturing:

Digital technologies are becoming more and more essential to productivity and effectiveness in industrial facilities and throughout supply chains. By substituting cables with wireless connection in factories, 5G may release machines from being tied to a certain place and provide producers with a far higher level of freedom [5]. Here the focus is on the use case where 5G technologies might increase the industrial sector's contribution to the global economy. But significant gains are also projected from other uses, such as those from supply chain integration, preventive maintenance, and production optimization, among others.

Autonomous robots and vehicles in industrial settings Industrial: Businesses will be able to reap the total rewards of the Fourth Industrial Revolution thanks to 5G-powered IoT applications. For instance, 5G will make it possible for robots and self-driving cars to navigate safely around facilities without the need for permanent infrastructure by using AI and laser-based awareness. The autonomous vehicles will cooperate with chargers, software, and services to function as an integrated inventory movement and robotics platform inside industrial centres and provide intelligent, safe, efficient, and dependable transportation. As a result, less labour will be required for manufacturing operations. By 2030, 5G industrial applications will increase the global GDP by US\$134 billion. As can be seen in Figure 2.13, China and USA are expected to get the biggest upliftment from 5G usage in this sector followed by Germany, Japan and UK.

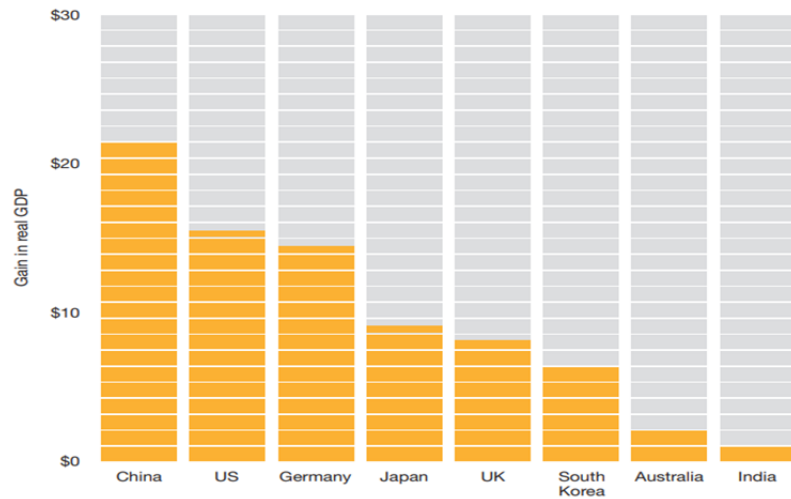


Figure 2.13: Country-wise GDP gain(US\$bn, 2019 values) from 5G in Industrial sector by 2030 [5]

2.4.1.4 5G in Financial Services:

This quick change in the digital landscape, coupled with the arrival of 5G and enhanced telepresence, offers firms the chance to redesign their customer strategy and journey

around more meaningful virtual experiences, allowing them to cut back on or do away with pricey physical channels like bank branch networks [5]. By 2030, financial services 5G apps will boost the global economy by US\$85 billion. By the end of the decade, 5G will have mostly enhanced customer experience and decreased losses from fraud in the banking industry.

2.4.2 Challenges

All these visions, benefits and predictions sound amazing. But challenges(a lot of them) persist that need to be overcome to reach the vision. The claims are extravagant, and the stakes are high. We'll now look at some of the major challenges that might prevent 5G from achieving its full potential.

a. Standardization: Lack of Standardization still remains the most challenging issue for 5G networks. Technology Standards and Regulatory Standards still need to be worked out for smooth implementation of 5G. Regarding technology standards, we refer to wireless communication, network protocols, data aggregation and unstructured data. When we talk about regulatory standards, we talk about encryption, data privacy, and security.

b. Energy optimization: In order to serve a large number of users, operate at high frequencies, support high data speeds, and build a lot of base stations, the implementation of 5G and beyond networks seek to be efficient. However, in order to accomplish these aims, a lot of energy is needed. The challenge for 5G networks is to optimize energy consumption while maintaining the promised level of service, which calls for practical and useful solutions.

c. Interferences: Interference is one of the most well-known difficulties faced by devices, and it becomes worse as the network evolves. The terminal can receive interference from multiple players, including other terminals, macro cells base stations, small cells base stations, relays, and other network nodes. As a result, effective and trustworthy interference detection, control, and cancellation methods are greatly needed [4].

d. Costly: The expensive infrastructure needed to support the network is one of the main obstacles to 5G implementation. For high-speed, low-latency communication, 5G technology needs a dense network of small relay cells and fiber-optic cables. This necessitates major financial investment on the part of telecom firms to upgrade their networks, which would not only be costly but also time-consuming. Additionally, deploying 5G infrastructure requires cooperation with local governments to secure permits and access to public infrastructure such as streetlights and utility poles. This can lead to delays and additional costs as companies navigate local regulations and negotiate access agreements.

e. Radio Spectrum: The limited availability of high-frequency spectrum is a challenge for 5G deployment, as it requires telecom companies to secure access to a significant amount of spectrum to support the network. However, the available spectrum is already heavily used by existing wireless technologies, such as 4G LTE and Wi-Fi, making it difficult for 5G to access the required bandwidth. The spectrum sensing technologies currently in use limit the ability to find open channels in high-frequency bands. Using

compressive spectrum sensing methods, processing time and computational complexity may be decreased [25]. However, more work must be put towards figuring out how to get over the restrictions of spectrum sensing at higher frequencies. As a result, effective allocation and aggregation algorithms are needed to divide the radio spectrum among SUs according to their demands.

f. QoS: As every application and service demands a different QoS, it becomes very challenging to provide high QoS to everyone and everywhere while yet meeting complete connectivity requirements. To ensure that 5G networks can provide dependable and consistent performance for a variety of devices and applications, smart and sophisticated procedures must be rethought.

g. Security: A "totally connected world" will also be especially susceptible to all security issues like cyberattacks, security loopholes, indirect tampering, disclosure of information, unfair use of resources, privacy leakage, ransomware, malware, crypto-jacking, identity theft, data breaches and many more. Machine learning algorithms work by collecting data from the environment to be processed and trained to produce intelligent actions to categorize or predict. By gaining access to one of the machine learning processes, supplying fictitious datasets for learning, or altering the sent information to produce false results, malicious individuals might identify security flaws and launch attacks.

h. Environmental Concerns and Hazards: With the vision of 5G for more and more device connectivity also implies more energy consumption hence more CO₂ emissions resulting in environmental concerns. Thus developing "efficient" and "safe" energy techniques in terms of production, consumption, and storage is- need of the hour!

** There is a widespread misperception that radiation-related health issues like cancer are caused by 5G technology. However, there is no scientific proof to back up this assertion. Due to the fact that 5G technology employs higher frequency bands than earlier wireless technologies, there are worries regarding the possible negative consequences of electromagnetic radiation exposure on human health. However, the amounts of electromagnetic radiation released by 5G technology are well below the safety thresholds established by regulatory bodies like the Federal Communications Commission(FCC) and the International Commission on Non-Ionizing Radiation Protection (ICNIRP).

2.4.3 5G, will you be able to keep your promises?

Definitely, it has the potential to fulfil its promises. But these discussed challenges need to get overcome or at least most of them. It would depend on a variety of factors like deployment of the necessary infrastructure, availability of "Compatible Devices", significant investments and importantly collaboration between the Government and Private Sector. Table 2.3 drafts what 5G has already achieved and what it still needs to consider.

It has already been started to be deployed in many countries. Initial tests have shown that it can deliver faster speeds, lower latency, and greater network capacity. However, to fully realize the benefits of 5G, significant investment is required in infrastructure, including the installation of small cells and fiber optic cables to provide widespread coverage. With

Table 2.3: What 5G already achieved and what it still needs to consider

What 5G already achieved	What 5G still needs to consider
Faster speed(up to 20 Gbps)	Coverage(especially in rural areas)
Lower latency(as low as 1 millisecond)	Interoperability
Increased Capacity	Enhanced Security
Improved network efficiency	Affordability

continued investment and innovation, we can be optimistic that 5G could potentially revolutionize the way we communicate and interact with technology.

2.5 Discussion

In the Discussion section, this report lists different policy approaches for policymakers, and a few interesting areas for future 5G research and directions have been pointed out.

2.5.1 Policy Principles for Global Policymakers

There are four critical areas where robust policy approaches and government actions are crucial as policymakers work to encourage the adoption of 5G: Deployment and spectrum access, innovation and investment, security, and standards. Because 5G cannot be implemented successfully otherwise, this report would advise policymakers to adopt a comprehensive strategy and take into consideration measures that take into account the concepts from every field.

a. Innovation and Investment: A supportive environment for investment and innovation must be ensured through 5G policies. Incentives for business and public sector investments in 5G research and development (R&D), investments in workforce development, and ensuring the unrestricted flow of data across borders—all these must be taken into account.

b. Enabling 5G Deployment and Access to Spectrum: Governments should cooperate to release spectrum, which is sometimes referred to as the "lifeblood" of wireless networks, and take measures to speed up 5G adoption [12]. Global 5G policymakers should prioritize making more spectrum available for 5G, promoting internationally harmonized spectrum bands, and avoiding inflating the price of 5G spectrum (for example, by setting high auction reserve prices) or allocating spectrum (for example, for vertical industries) that has been designated for mobile communication.

c. Security: As 5G's scale increases, so will its dangers and concerns. The security of 5G networks needs to be prioritized by Government policymakers in an appropriate manner. Policies should address concrete, identifiable security risks and have possible solutions to such circumstances so that in case such a security breach happens in future, they

could already have a proposed solution at hand. Government and private stakeholders must share responsibility and collaborate to mitigate cybersecurity threats to network infrastructures.

d. Standards: Standards for 5G must be industry-led, and competition must be promoted. Competition leads to improved standards, drives innovation and allows for optimal market-based choices. Policymakers should promote internationally harmonized 5G standards or technical specifications. (and not country-specific standards).

2.5.2 Open Issues, Future Directions and Motivation for Research

Some open issues that can be considered for further research and motivation are:

a) Fiber ring network protection: Simply explained, protection in fiber network means the availability of an alternative path in case of failure. A typical example can be considered with the 4:1 setup where 1 redundant data path is available for 4 actual data transmitting paths. In case of a failover in any of the 4 actual paths, the redundant one is switched on so that data transmission is not hampered. Interconnected protected rings with ADM/DCX is considered the best protection technique in fiber network by many authors. This fiber ring technique can be used for smooth 5G traffic.

b) Synchronous Digital Hierarchy(SDH): SDH is a standardized protocol for high-speed networks and connections. Although it is a bit old-schooled but still 5G can learn the firm SDH architecture which provides availability of high-speed connectivity and good resilience to network outages.

c) Redesign of encryption methods: For 5G services, encryption techniques must be developed in a manner that can provide strong security and privacy protection for user data and conversations. Here are a few possible strategies for updating encryption techniques for 5G services: i)Post-Quantum Cryptography: As quantum computing develops, conventional encryption techniques are becoming more open to intrusion. A new encryption technique called post-quantum cryptography makes use of mathematical formulas that are thought to be impervious to assaults from quantum computers. ii)Post-quantum cryptography may improve security against potential assaults in 5G networks. Hybrid Encryption: To provide strong protection for data in transit, hybrid encryption combines the advantages of symmetric and asymmetric encryption. In a hybrid encryption technique, the symmetric key is encrypted using asymmetric encryption, while the data is encrypted using symmetric encryption. This strategy may provide improved security and quicker encryption and decryption times. iii)Multi-factor authentication (MFA): MFA is a security measure that prevents users from accessing their accounts without first providing two or more pieces of identification. In order to offer an extra layer of protection and guard against illegal access, MFA may be deployed in 5G networks. iv)Zero-Trust Security: This method is based on the idea that no person or device should ever be automatically trusted. Instead, a tight set of regulations and authentication procedures govern how access to resources is allowed. Zero-trust security may assist in safeguarding against insider attacks and unwanted access in 5G networks.

d) Network Slicing, Network Caching: Further improvement for Network Slicing can be to dynamically allocate devices belonging to a particular slice to a different slice in case their resource requirement changes so that resource utilization can be maximised. In order to make the most use of the base station storage, operators may anticipate the resources needed based on the future content of the users. Operators may predict users' demands for resources and upcoming content via the network slices with 5G.

e) Massive MIMO: For massive MIMO, examples of future directions include: i) using new air interfaces, ii) and new MIMO schemes to optimize the high frequencies and latency

Despite the suggested solutions, there are still a number of research gaps and issues arising from the new technologies and use cases for 5G networks.

2.5.3 6G and Beyond!

Wireless network w/o "limitations"- 6G can be characterized by: a) Streaming w/o buffering, b) more swift real-time, online gaming across the world, c) Almost "no" delay: ms and below to μ /s, d) Pervasive augmented reality (Holograms, smart material) e) Terabit/s transmission speeds along with the availability of smart antennas.

Requirements [6]: Large internal memory will be required, particularly in mobile devices, high-speed capacity and fiber optics backbone network(ex-PON based). Commercial availability of 6G projected for the 2030s.

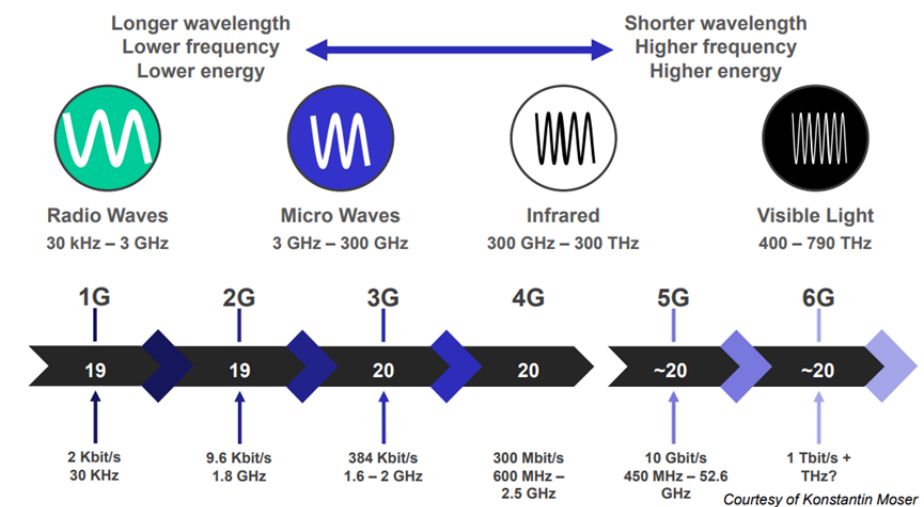


Figure 2.14: Evolution of Bandwidth/frequency use [6]

2.5.4 Intriguing 5G Questionnaire Set

The final part of this report would contain some interesting questions regarding views of 5G from a different perspective. The answers are equally intriguing and the opinions might vary from reader to reader.

2.5.4.1 Pay more, Play more!

Consumers are wising up to the benefits of 5G—but will they pay for it? According to the GSMA Intelligence Consumers insights survey 2019, globally only 2 percent of the respondents who intend to upgrade to 5G are willing to pay more. (Figure 2.15) 14 percent of them are willing to pay up to 20 percent. But 41 percent are willing to pay only up to 10 percent whereas 39 percent do not want to pay extra for upgrading to 5G.

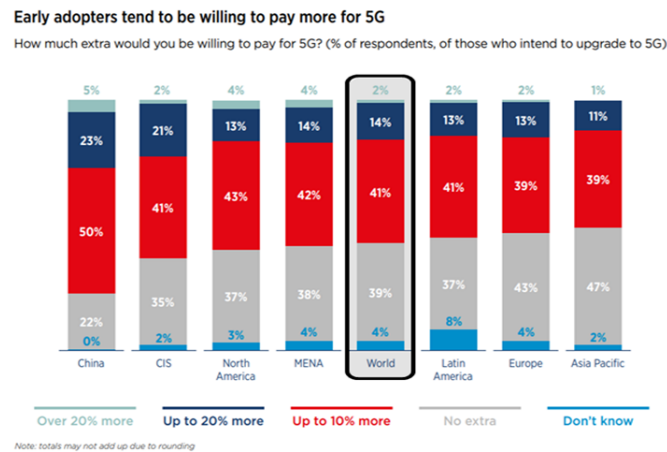


Figure 2.15: How much extra would you be willing to pay for 5G? [2]



Figure 2.16: Awareness of 5G [2]

Figure 2.16 shows that countries like Russia, Chile, and Argentina do not seem to be highly aware of 5G to the fullest but still show high interest in the upgradation. China, South Korea and the USA are the early enthusiasts whereas for many European countries either 5G is still not on the radar or they are not yet convinced of its potential.

2.5.4.2 Money making over quality service?

Will the Operators be more focused on revenue making than customer satisfaction? As can be seen in Figure 2.17, operators will likely be focused on generating revenue to recoup infrastructure and tech costs. But customer satisfaction is also important for retaining customers and attracting new ones. Ultimately, the success of operators in the 5G market will depend on their ability to balance revenue generation with customer satisfaction.

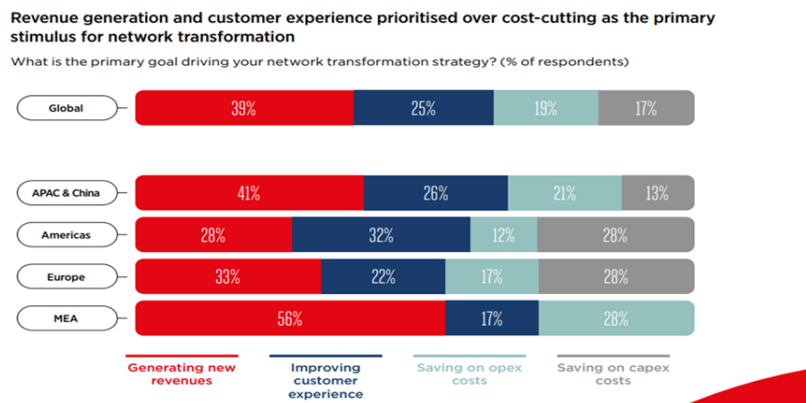


Figure 2.17: Priority list for 5G Operators [2]

2.5.4.3 More Jobs or Jobless More?

How will the promises of the 5G network affect the job market, particularly in the tech industry and related fields? Particularly in the tech industry, it is expected that there will be increased demand for skilled workers. Although ChatGPT and other AI chatbots are unfolding wonders, skilled developers will always be on the demand. However, there might be disruption of existing job roles and increased competition for jobs. Also, the demand for remote work is expected to increase.

2.5.4.4 5G for Good or Bad?

What would be the impact of 5G Network on Social life, Learning, and Addiction (for games or social media)? Like both sides of the coin, any innovation also goes both ways. The students' want for social media accessibility rose as a result of anticipated 5G mobile

broadband [14]. Figure 2.18 shows a Zambian study on the effects of social media use at night. Crucially, the survey revealed that 66.8% of respondents, the majority of whom were adolescent students, reported having trouble falling asleep at night due to a fear of missing out. Other issues reported were experiencing headaches or pain in the eyes, phone screens disturbing them during bedtime, dose in class due to late night social media use etc. E-learning though would have huge scope for improvement.

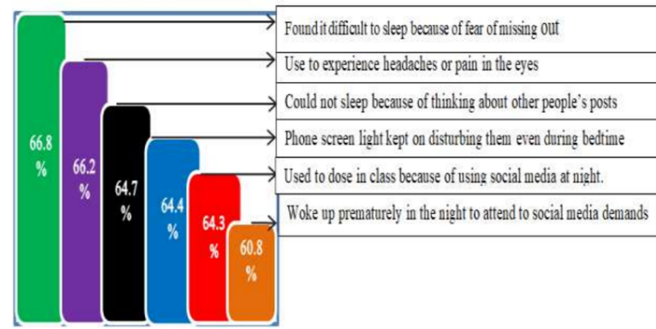


Figure 2.18: Effects of social media use at night [14]

2.6 Conclusion

A brief summary of what was seen in this report- starting from the comparisons of different Generations with 5G it looked at the Contribution Prediction of 5G in different sectors. Looked at different 5G enablers and MEMS timing devices of 5G followed by its Vision and Challenges which raised the reality check question- "Will 5G keep its promise?" We then raised policy concerns regarding different challenging open issues and possible future directions for further research and development.

These challenging open issues are not something new. Any innovation (technical or non-technical) has always witnessed challenges. It is necessary to enhance the architecture with security as a future priority. Although that might result in network performance reduction, necessitating effective and cutting-edge solutions. Continued investment and innovation and significant collaboration between Government and Private sectors would be the determiners of the success of 5G by the end of the decade.

On an ending note, praising the exploration tendency of human nature which has made every single innovation possible. It has existed since time immemorial and should also never cease to exist. It is because of this innate desire of humans to explore and discover new things that have made us humans superior to all other creatures at least on this planet. As technology and trends are constantly evolving what may seem like a luxury or a novelty today may become a general use case or a standard in the future. And correctly as they say SKY IS THE LIMIT!

Bibliography

- [1] Statista Research Department: 5G - Statistics & Facts; URL: <https://www.statista.com/topics/3447/5g/#topi-c0verview>, Last visit April 6, 2023
- [2] GSMA HEAD OFFICE: The mobile economy 2020; URL: https://www.gsma.com/mobileeconomy/wp-content/uploads/2020/03/GSMA_MobileEconomy2020_Global.pdf, Last visit April 6, 2023
- [3] Ahmad, Wan Siti Halimatul Munirah Wan, Nurul Asyikin Mohamed Radzi, F. S. Samidi, Aiman Ismail, Fainz Abdullah, Md Zaini Jamaludin, MohdNasim Zakaria. "5G technology: Towards dynamic spectrum sharing using cognitive radio networks." IEEE access 8 (2020): 14460-14488.
- [4] Salahdine, Fatima, Tao Han, Ning Zhang. "5G, 6G, and Beyond: Recent advances and future challenges." Annals of Telecommunications (2023): 1-25.
- [5] PWC: The global economic impact of 5G; URL: <https://www.pwc.com/gx/en/tmt/5g/global-economic-impact5g.pdf>, Last visit April 6, 2023.
- [6] "Wireless Telecommunication Systems": Burkhard Stiller and Jochen Schiller, FU Berlin, 2022; URL: http://www.csg.uzh.ch/teaching/ss05/mobsys_v/lectures/M04.pdf
- [7] Unlocking the Promise of 5G - Ash Bhalgat; URL: <https://developer.nvidia.com/blog/unlocking-the-promise-of-5g>, Last visit April 6, 2023
- [8] Delivering On The Promise Of 5G, Why Should You Rethink Timing?: Markus Lutz; URL: <https://www.forbes.com/sites/forbestechcouncil/2020/12/15/delivering-on-the-promise-of-5g-why-should-you-rethink-timing>, Last visit April 6, 2023
- [9] The terrifying potential of the 5G Network: By Sue Halpern, 26 April 2019; URL: <https://www.newyorker.com/news/annals-of-communications/the-terrifying-potential-of-the-5g-network>, Last visit April 6, 2023
- [10] Kinan Ghanem , Stephen Ugwuanyi , Rameez Asif , James Irvine. "Challenges and Promises of 5G for Smart Grid Teleprotection Applications"; Conference Location: Dubai, United Arab Emirates; 02 November 2021; URL: <https://ieeexplore.ieee.org/document/9615649>

- [11] Niranjana Lal, Shobhit Mani Tiwari, Devbrat Khare and Megha Saxena. “Prospects for Handling 5G Network Security: Challenges, Recommendations and Future Directions”; 2021; URL: <https://iopscience.iop.org/article/10.1088/1742-6596/1714/1/012052/pdf>
- [12] ITI’s 5G Policy Principles and 5G Essentials for Global Policymakers; URL: https://www.itic.org/policy/ITI_5G_Full_Report.pdf, Last visit April 6, 2023
- [13] Sheshadri Chatterjee, Ranjan Chaudhuri, Alkis Thrassou, Demetris Vrontis. “Social network games (SNGs) addiction: Psychological dimensions and impacts on life quality and society”; Volume 177, April 2022; URL: <https://www.sciencedirect.com/science/article/pii/S0040162522000610>
- [14] Akakandelwa AkakandelwaHarry, Jordan Silomba. “The impact of bed-time compulsive social media use on students’ quality of sleep in the copperbelt colleges of education, Zambia”; March 2021; URL: https://www.researchgate.net/publication/350029921_THE_IMPACT_OF_BEDTIME_COMPULSIVE_SOCIAL_MEDIA_USE_ON_STUDENTS_QUALITY_OF_SLEEP_IN_THE_COPPERBELT_COLLEGES_OF_EDUCATION_ZAMBIA
- [15] Introduction to Wireless and Telecommunications- ASM Technologies; URL: <https://www.asmltd.com/introduction-wireless-telecommunication/#:~:text=Wireless%20communication%20technology%20transmits%20information,and%204G%20networks%2C%20and%20Bluetooth%20;Last%20visit%20April%206%2C%202023>
- [16] 5G network architecture | 5G Architecture- RF Wireless World; URL: <https://www.rfwireless-world.com/Tutorials/5G-network-architecture.html#:~:text=5G%20Protocol%20Stack%20&text=0WA%20Layer%3A%200WA%20layer%20is,lower%20and%20upper%20network%20layers;Last%20visit%20April%206%2C%202023>
- [17] R. I. Rony, A. Jain, E. Lopez-Aguilera, E. Garcia-Villegas, and I. Demirkol. “Joint access-backhaul perspective on mobility management in 5G networks”; in Proc. IEEE Conf. Standards Commun. Netw. (CSCN), Sep. 2017, pp. 115–120
- [18] Mousa SH, Ismail M, Nordin R, Abdullah NF (2020) “Effective wide spectrum sharing techniques relying on CR technology toward 5G: a survey.” J Commun 15(2):122–147
- [19] Salahdine F, El Ghazi H (2017) “A real time spectrum scanning technique based on compressive sensing for cognitive radio networks. In: 2017 IEEE 8th annual ubiquitous computing, electronics and mobile communication conference”; UEMCON 2017, vol 2018-Janua, pp 506–511
- [20] Salahdine F, Kaabouch N, El Ghazi H (2016). “A survey on compressive sensing techniques for cognitive radio networks”; Phys Commun 20:61–73
- [21] Salahdine F (2018) “Compressive spectrum sensing for cognitive radio networks”; arXiv:1802.03674
- [22] Sun S, Gong L, Rong B, Lu K (2015) “An intelligent sdn framework for 5g heterogeneous networks.”; IEEE Commun Mag 53(11):142–147

- [23] Y. Niu, Y. Li, D. Jin, L. Su, and A. V. Vasilakos. “A survey of millimeter wave communications (mmWave) for 5G: Opportunities and challenges”; *Wireless Netw*, vol. 21, no. 8, pp. 2657–2676, Nov. 2015.
- [24] Network Slicing–TechTarget; URL: <https://www.techtarget.com/whatis/definition/network-slicing>, Last visit April 6, 2023
- [25] Salahdine F, Kaabouch N, El Ghazi H (2016): “A survey on compressive sensing techniques for cognitive radio networks”; *Phys Commun* 20:61–73

Chapter 3

SCION: A Clean-slate Internet Architecture

Elliott Wallace

The current Internet architecture faces challenges in terms of security, scalability, transparency, and control. In its original design, these factors were not necessarily considered critical. To overcome these limitations, various clean-slate approaches have been proposed. This seminar report focuses on SCION (Scalability, Control, and Isolation On Next-Generation Networks), a clean-slate future Internet architecture designed to provide high availability and efficient packet delivery. Simply put, SCION's approach is to revolutionize inter-domain routing by replacing the Border Gateway Protocol (BGP). The intention is to increase scalability by removing the need for stateful routers, thereby mitigating the effect of ever-growing inter-domain routing tables. It likewise introduces security mechanisms to solve BGP's vulnerability to attacks such as prefix hijackings. SCION provides transparency and control over network communication paths enabled by its path-aware routing capabilities. Coupled with its multipath communication features and hidden path segments, this also makes the architecture more resilient against distributed denial of service attacks. Compared to other future Internet proposals, SCION is mature and advanced in terms of industry adoption. This report aims to analyze the general applicability of a clean-slate approach to the Internet's architecture. As SCION has already been deployed in different productive settings, such as the financial industry, it is a suitable candidate for this purpose. In conclusion, SCION represents a promising alternative to the current Internet architecture that has the potential to enhance the security, reliability, and efficiency of end-to-end communication. The coexistence capabilities of SCION with the current IP architecture allow for a gradual transition. However, the feasibility of widespread adoption at the consumer level requires further research and testing.

Contents

3.1	Introduction	51
3.2	Background	51
3.2.1	Current Internet Architecture	51
3.2.2	Destination-based Packet Routing	53
3.2.3	Trust Models	54
3.3	The Necessity of a Clean-slate Approach	54
3.3.1	Scalability Issues	54
3.3.2	Security Issues	55
3.3.3	Lack of Transparency and Control	56
3.4	SCION Architecture	57
3.4.1	SCION Objectives	57
3.4.2	Architecture Components	58
3.4.3	Control Plane	59
3.4.4	Data Plane	61
3.4.5	Security	61
3.4.6	Deployment	62
3.5	Real World Applicability of SCION	62
3.5.1	Industry adoption of SCION	63
3.5.2	Benefits and challenges of SCION adoption	63
3.5.3	Comparison with other clean-slate approaches	65
3.6	Discussion and Summary	65

3.1 Introduction

The Internet is one of the last century's most revolutionary and influential inventions. Since its inception in the 1960s, it has grown exponentially in many ways and has become an integral part of today's society. Playing a major role in our everyday lives, the Internet enables communication and collaboration across the globe, promoting and transforming education, entertainment, and business. However, the current Internet architecture faces certain challenges that threaten its perspectives for the future, such as security and scalability problems. The Internet was not designed with these requirements in mind and rather focused on creating a network of collaborating networks based on mutual trust, following a different set of design goals and principles [35]. Despite the enormous growth of the global Internet infrastructure and changes in use cases and applications, the fundamental protocols it was built upon remained largely the same. Changing an existing system by building on it can prove difficult. As a result, the Internet is vulnerable to various attacks such as prefix hijacking or denial of service (DOS). Therefore, a number of clean-slate approaches have been proposed to redesign the Internet architecture from scratch. One of these approaches is SCION (Scalability, Control, and Isolation On Next-Generation Networks) [1], a modern Future Internet architecture that aims to offer high availability and efficient point-to-point packet delivery.

This seminar report aims to describe the main components and features of the SCION architecture, evaluate its benefits and challenges, and use the example of SCION to discuss the real-world applicability of a clean-slate approach to the Internet's architecture. The report is structured as follows: Section 3.2 provides some background on the current Internet architecture and its routing system, as well as how it models trust. In section 3.3, the need for a clean slate approach is justified by discussing scalability and security issues, in addition to shortcomings in transparency and control. Section 3.4 takes a closer look at the SCION Internet architecture, describing its objectives, components, its control, and data plane mechanisms, followed by security and deployment aspects. Moreover, section 3.5 elaborates on the real-world applicability of SCION, first looking at its adoption in the industry. The benefits and challenges of adopting SCION are evaluated and a comparison with other clean-slate approaches is attempted, followed by a discussion 3.6 of the applicability of the clean-slate approach.

3.2 Background

The following section is intended to provide some context regarding the status quo of the Internet, considering its architecture, routing infrastructure, and trust models.

3.2.1 Current Internet Architecture

A network is essentially a set of interconnected end systems or hosts and intermediate systems, such as routers and switches [3]. In relation to the Internet, such a network (or

group of smaller networks) that is independent and controlled by a single administrative entity is called an Autonomous System (AS) [2]. Following this, the Internet as we know it today can be described as a collection of interconnected autonomous systems, resulting in a global decentralized network that is partitioned into different administrative domains [3, 5]. Although the connections between ASes follow no explicit, imperative structure, which is due to the fact that the Internet is decentralized, a hierarchy with different levels of ASes can be observed. These categories correspond roughly to the size of the respective ASes and the number of end hosts they can reach[5]. Smaller domains, so-called stub-ASes are formed by academic networks or smaller regional Internet Service Providers (ISPs). Larger ASes provide transit services to smaller ASes by forwarding traffic to destinations in remote domains. As an example, a campus-wide academic stub-AS may connect to a national ISP, which can be a tier 2 AS, for transit services. The national tier 2 AS in turn acquires transit from a tier 1 AS. Different tier 1 ASes connect with each other through peering relationships, building the backbone of the Internet that enables global connectivity [4].

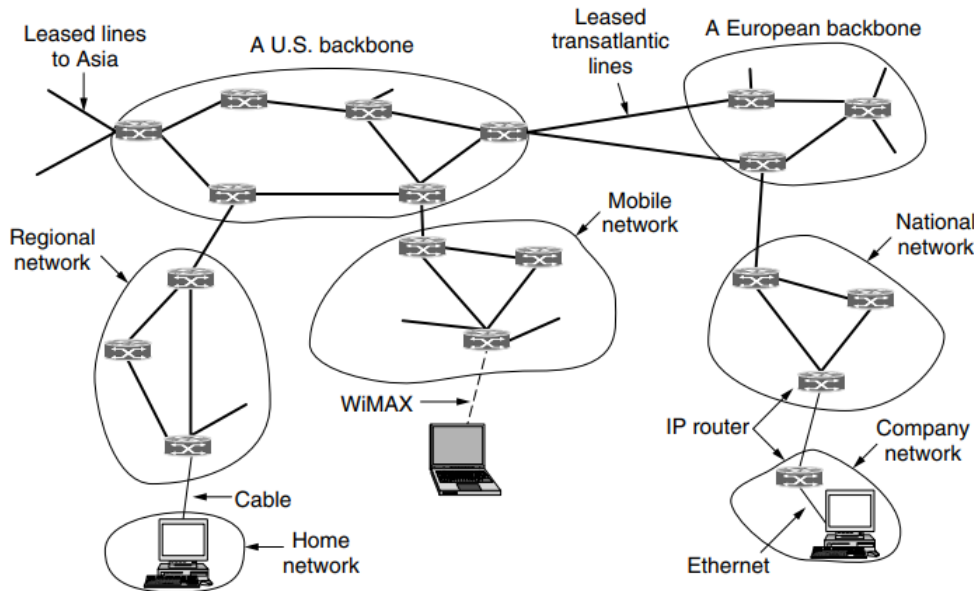


Figure 3.1: The Internet's structure. From [4], Figure 5-46, p. 443

On the network layer, the Internet comprises routers that interconnect networks through the Internet Protocol (IP) [7], which provides a best-effort way of delivering packets from source to destination. The IP provides a universal way of addressing entities on the Internet via IP addresses. An AS is associated with one or more contiguous blocks of IP addresses, called IP prefixes, that make up individual networks. Simplified, communication on the Internet can be modeled with the following steps: At the source host, the transport layer data stream that is to be sent is broken down into packets, containing the destination's IP address, and handed to the network layer. The packets are then forwarded from one IP router to the next, based on the target address, until they can be delivered to the destination host, where the packets can again be handed to the transport layer. Given that there are usually many possible paths a packet can take from a source to a destination, there has to be a mechanism to determine which path to choose. This

task is executed by routing policies [4].

3.2.2 Destination-based Packet Routing

The Internet Protocol provides a universal packet format and addressing scheme [7]. For information transmission, IP networks encapsulate data into packets that carry the IP address of the desired destination host in their header. On an interconnected network, such as the Internet, there may be many different possible paths a packet can travel through, moving from a source to its destination. Routing is the process of selecting the path a packet will effectively take.

IP networks rely on routing protocols to construct a distributed routing state, enabling communication between hosts via links and routers that interconnect these networks. Routing protocols define standards for intermediate systems (routers) to exchange information about reachable addresses and enable the dynamic selection of paths [6]. Routing is a decentralized process, where packets are forwarded from one router to the next until they reach their destination. The decision of where a packet is forwarded to is done individually at each intermediate router. Therefore, every router needs to maintain some state in the form of a routing table. Each entry in such a table connects a destination address to the next router along the path, also known as the next-hop router [8].

Modern router architectures divide their functionalities into a control plane and a data plane [4]. The control plane is the software layer, applying the different routing protocols. It is responsible for building the routing table. Routers use the information gathered in the routing table to compile a forwarding table that is stored in memory. The domain of the hardware-based data plane is to perform lookups on the forwarding table upon receiving an incoming packet. This lookup operation entails a longest-prefix matching process, based on the destination address in the packet's header. It results in the address of the hop, where the packet is subsequently forwarded to. In order to make this process efficient, the memory that holds the forwarding tables needs to support fast lookup operations [8].

The Internet's routing algorithm operates on two levels. Intra-domain routing entails the routing inside autonomous systems, whereas inter-domain routing refers to the routing between different ASes. The protocols used in intra-domain routing are called Interior Gateway Protocols, with examples such as Open Shortest Path First (OSPF) [9] and Intermediate System to Intermediate System (IS-IS) [10], which are both link-state routing protocols. For inter-domain routing, the de facto standard protocol of today's Internet is the Border Gateway Protocol (BGP) [2].

The inter-domain topology of the Internet can be seen as an undirected graph. The nodes are given by the different routing domains and the edges between the nodes represent the interconnections between these ASes, where routing information and IP traffic are exchanged. The Border Gateway Protocol is the means by which routes are exchanged in this context [5]. The specification of version four of the BGP protocol specifies: "The primary function of a BGP speaking system is to exchange network reachability information with other BGP systems", where a BGP speaking system (or BGP speaker) refers to a router that implements the BGP protocol [2]. A BGP speaker uses BGP UPDATE messages

to incrementally announce new routes or to withdraw invalid ones to neighboring ASes. BGP is a path-vector protocol. Therefore, when advertising a new route, each AS adds its ID number to the path attribute on the `UPDATE` message before propagating it further. `UPDATE` messages contain a number of important attributes, such as local preference, AS path length, and origin type, that are used to implement specific policies for AS-level routing. Those routing policies are used to model business and organizational relationships between different ASes, which are usually defined in service level agreements (SLAs) or peering contracts. Ultimately, the policies configured in BGP routers are also used for filtering the incoming routes to decide which BGP route is used for each IP prefix and also to determine which routes are propagated further to neighboring ASes.

3.2.3 Trust Models

The current Internet infrastructure relies on a variety of mechanisms and protocols to establish trust among different entities, ensuring the reliability and security of communications. This section gives a concise introduction to the Internet's trust models, focusing on Public Key Infrastructure (PKI) and Transport Layer Security (TLS).

A Public Key Infrastructure (PKI) is a system that enables authentication and encryption, using public key cryptography. The system is responsible for the management of digital certificates. A certificate represents a binding of a public key to an identity or domain name on the Internet. Certificates are issued, revoked, and signed by trusted third parties, called Certificate Authorities (CAs). These are usually organized in a hierarchical manner. A certificate chain is a sequence of certificates that link any given certificate with the root CA certificate [36].

One of the most widely used security protocols on the Internet is the Transport Layer Security (TLS) [21] protocol, as it is used to secure HTTPS connections. TLS is used to establish secure communication between a client and a server. It relies on certificates that are exchanged upon the initial connection setup to verify the identities of the communicating parties.

3.3 The Necessity of a Clean-slate Approach

Although the current Internet architecture has lasted for a considerable time and enabled various applications, it faces several challenges, such as scalability, security, and transparency, that could be addressed by a comprehensive redesign. The following section discusses these issues and reasons that speak for a clean-slate approach to the Internet's architecture.

3.3.1 Scalability Issues

As more and more people and devices around the world gain access to the Internet, the number of IP addresses steadily increases. This growth directly impacts the size of inter-

domain routing tables. As of January 2023, the number of entries in BGP routing tables amounted to about 940'000 IPv4 prefixes and 172'400 IPv6 prefixes [11]. In light of this development, it is reasonable to question the scalability of the current inter-domain routing system.

The creators of SCION [1] identify a lack of scalability in the current Internet's architecture, especially in relation to inter-domain routing and the BGP protocol. One of the reasons they name is the use of stateful routers, where for each incoming packet the forwarding table has to be queried before the packet is forwarded to the next-hop router. For efficient packet forwarding, these lookup operations must be accordingly fast. Router architectures typically rely on highly performant hardware in the form of ternary content-addressable memory (TCAM) to meet this requirement. Increasing the size of routers' TCAM memory to accommodate the space requirements of the growing forwarding tables has significant drawbacks, as the hardware is expensive and very demanding in terms of energy usage [13]. The authors in [1] state that routing tables have reached their limits in terms of scalability, due to prefix de-aggregation and multihoming.

In [1], it further mentions the fact that the overhead for BGP routing, measured by the amount of BGP `UPDATE` messages sent, is proportional to the number of prefixes registered. The paper states that this leads to increased convergence times until a stable routing state is reinstated. Convergence is a process that is triggered by changes in the topology of a network, be it by a new prefix being announced or due to a link failure. BGP speakers exchange `UPDATE` messages until a new stable state of the routing table is achieved again [14]. During the time it takes for a network to converge, instabilities occur in the system.

Given the current situation regarding energy prices, especially in Western countries, and considering the climate implications power-hungry BGP routers have, the scalability of the current routing infrastructure may in fact be questionable. The assertion that inter-domain routing with BGP has reached its limits in terms of scalability [1] has not yet been proven to be right, as the BGP routing infrastructure has not yet collapsed nor is it expected to do so anytime soon [11]. Slow convergence in turn is a well-known issue of the BGP protocol. The path exploration that takes place during convergence phases after a link failure can lead to many BGP advertisements being exchanged until valid paths are discovered [15].

3.3.2 Security Issues

The current Internet architecture is based on a set of protocols, that were designed decades ago. Security was not a primary concern back then, as the focus was more on providing basic functionality for data transmission and routing. Requirements that are considered very important nowadays, like confidentiality, integrity, and authenticity, were not considered critical in the early times of the Internet. As a result, the Internet is vulnerable to various attacks and exploits [38].

Even though being a cornerstone of the Internet, the Internet Protocol does not provide inherent authentication or integrity mechanisms. This for example allows the spoofing

of IP packets, which can lead to attacks such as Denial-of-Service (DoS) being harder to trace. IP doesn't have confidentiality or encryption features out of the box, exposing traffic to eavesdropping or interception. To address these limitations, several extensions have been proposed, such as IPsec, which provides security at the network layer. However, it is not widely deployed, except in Virtual Private Networks (VPNs) [20].

Securing end-to-end communications has been implemented at the transport layer in the form of the Transport Layer Security (TLS) protocol, which is widely used for securing web traffic and other applications on the Internet [21]. TLS relies on certificates and Certificate Authorities (CAs) in a PKI that is based on an oligopolistic trust model, in which any one of hundreds of trusted third parties can issue a certificate. This poses a problematic attack surface, as a CA can be compromised by an attacker who can then issue forged certificates for any domain or identity that trusts this CA [1].

Inter-domain routing also causes great security weaknesses in the security of the Internet. BGP, being the primary inter-AS routing protocol, does not have any built-in mechanics to verify the authenticity or validity of the routing information exchanged in BGP `UPDATE` messages [2]. This makes BGP susceptible to attacks, such as prefix hijacking. Prefix hijacking occurs when an AS advertises a route for a prefix that it does not own or have permission to use, with the goal of diverting traffic intended for the legitimate owner of the prefix [22, 15]. In 2008, a Pakistani ISP hijacked the prefix of YouTube and blocked access to the platform for several hours [23]. BGPSEC [24] is a protocol that has been proposed to address this, by adding cryptographic signatures to BGP messages to prevent unauthorized route announcements [24]. However, BGPSEC requires all ASes in a path to support the protocol, which creates a significant barrier to individual deployment and therefore widespread adoption. In contrast to TLS, BGPSEC only relies on a relatively small number of trust roots, which can also be seen as a drawback, as the whole system depends on the integrity of a few entities[1].

3.3.3 Lack of Transparency and Control

In the current Internet architecture, the transport layer's end-to-end functionality and the network path's properties on the layer below are separated [18]. TCP can be used to establish a reliable connection to a remote endpoint, and TLS enables the authentication of this entity, but the respective end-host has little information about the path carrying their traffic [19]. This path is determined autonomously by the Internet's control plane and is beyond the control of any individual endpoint. Once a packet has left its source, the source has no influence over the forwarding decisions taken at intermediate routers. The receiving host also has little control and information, as it cannot enforce specific path preferences nor verify the path taken [8]. This lack of transparency and control opens up several opportunities for malicious activities [8] and limits the flexibility of networking applications [19].

Further, path awareness is a missing property of the current Internet architecture. The Path Aware Networking Research Group (PANRG) [18] defines path-aware Internet as a network, "[...] in which endpoint discovery of path properties and endpoint selection of paths used by traffic exchanged by the endpoint are explicitly supported, regardless of the

specific design of the protocol features which enable this discovery and selection” [17]. This property enables protocol designers to add new capabilities to transport and application layer protocols. For instance, it could be used to increase failure resilience by applying disjoint network paths or to exclude paths that may be susceptible to eavesdropping attacks [19].

In addition to the increased control achieved through path awareness, a transparent and secure Internet architecture should provide a way to validate network paths. On the one hand, it must be possible to make sure a packet travels through the chosen path and, on the other, to check whether it has in fact taken that way. Therefore, path validation enhances packet delivery by implementing path enforcement. This ensures that intermediate routers forward packets as specified by the source and path verification, which lets destinations and intermediate nodes verify path integrity [8].

3.4 SCION Architecture

Clean-slate Internet architectures are network designs that do not inherit most of the existing Internet protocols or principles. They aim to overcome the fundamental limitations of the current Internet, such as security, scalability, and mobility, by essentially redesigning it from scratch. In contrast, evolutionary or incremental approaches try to improve the Internet by modifying or extending the existing architecture. The disadvantage of this approach is, that it is constrained by the existing protocols and design decisions [34]. A prime example of a clean-slate Internet architecture is SCION, which stands for Scalability, Control, and Isolation on Next-Generation Networks [1].

3.4.1 SCION Objectives

The main motivation behind SCION is to address the limitations and vulnerabilities of the current Internet architecture, such as the ones discussed in section 3.3. This section lists the main goals SCION wants to accomplish with its design.

Availability in the presence of adversaries: SCION aims to offer a network architecture that provides reliable communication, even if under attack. Malicious entities might try to drop or alter packets on their path or divert their intended route. Distributed denial-of-service (DDoS) attacks are becoming increasingly common. As critical applications and services, such as banking, healthcare, or emergency response depend on high availability at all times, it is SCION’s objective to provide connectivity as long as an attacker-free path exists between endpoints [1].

Transparency and control: In SCION, transparency and control encompass both forwarding paths and trust. In the context of paths, transparency refers to the ability of the sender as well as the receiver to discover and verify networking paths. Furthermore, senders should be able to choose the path their traffic takes and receivers should have control over the paths they can be reached through. SCION also aims

to enable every end host or user to have full oversight of the trust roots they have to rely upon for certificate validation. This transparency in turn provides them with control, in the sense that they can choose which trust roots they wish to accept [1].

Efficiency and scalability: The ability of a network to handle large amounts of traffic and network endpoints without compromising performance or quality of service is critical for supporting the growing demand for internet connectivity. SCION increases efficiency and scalability by avoiding the use of stateful routers. The architecture also supports multi-path communication, which can increase end-to-end bandwidth and enable dynamic traffic optimization [1].

3.4.2 Architecture Components

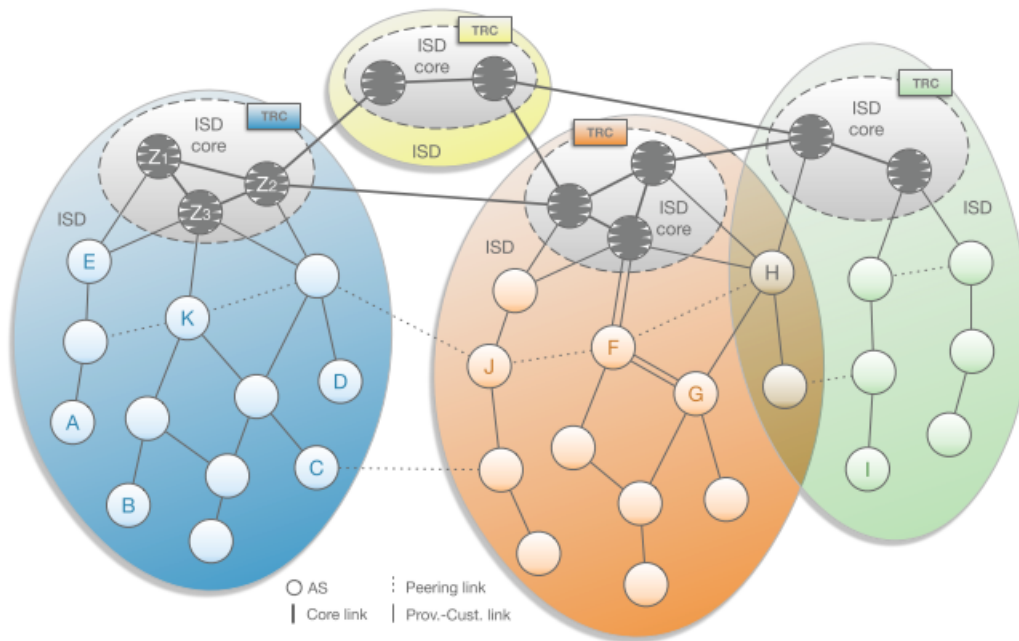


Figure 3.2: SCION Architecture. From [1], Figure 2.1, p. 18.

One of the key features of SCION is that it organizes autonomous systems (ASes) into groups of independent routing planes, called **Isolation Domains (ISDs)**. Figure 3.2 shows four distinct and interconnected ISDs. Typically, the ASes in an isolation domain share some common properties, like geographic location or legal jurisdiction. Every ISD contains an **ISD Core**, formed by one or more ASes, so-called core ASes, who together take over the administration of the ISD. The core also defines the central policy that governs an ISD, the **Trust Root Configuration (TRC)**. In any ISD, the TRC is a policy document that defines the roots of trust that are relied upon in this ISD for the validation of certificates. The roots of trust are technically a set of signed root certificates. They are used to sign the ISD's CA certificates, which are then used to sign the individual AS certificates. End hosts can use TRCs to validate bindings between names and public keys or addresses of entities within an ISD. To join an ISD, an AS must accept the ISD's TRC, which will typically

happen implicitly by purchasing connectivity from a provider within an ISD [1, 32]. In comparison, the current Internet architecture relies on a hierarchical PKI for certificate validation, based on a set of almost universally trusted CAs that can issue certificates for any domain name. However, this PKI model has several drawbacks, such as the lack of transparency and accountability of CAs.

In the context of ISDs, *isolation* refers to the separation of an ISD’s control plane from those of other ISDs. In their routing infrastructures, the scope of SCION addresses and routes is limited to their respective ISDs. In contrast to BGP, where updates are disseminated profusely, routing information between ASes in an ISD in SCION is not propagated beyond the ISD. To enable global connectivity, SCION uses a dedicated, secure inter-ISD routing protocol. The isolation property can also be observed in SCION’s authentication infrastructures, as the scope of any root key defined in a TRC is confined to its own ISD [1].

The exploration of routing paths in SCION is implemented through a *beaconing* process. It uses **path-segment construction beacons (PCBs)** that are propagated among the ASes within an ISD (intra-domain) or between core ASes of different ISDs (inter-domain) to discover path segments. While traversing a network, a PCB gathers AS-level path information. Eventually, this information is combined into paths traversing a series of ASes and encoded into the headers of SCION packets. This concept is known as **packet-carried forwarding state (PCFS)** [1].

A SCION AS features some key infrastructure components [1, 32]:

Beacon servers The beacon service is responsible for discovering path information. A beacon server generates, receives, and propagates PCBs, enabling the SCION beaconing process.

Path servers The path service stores information about discovered path segments. Path servers respond to queries by end-hosts for path information.

Name servers SCION name servers translate domain names into SCION addresses, analogous to DNS servers.

Certificate servers The certificate service is responsible for securing inter-AS communication. Certificate servers store cached copies of the TRC and manage keys and certificates. The beacon service queries the certificate service to validate PCBs before forwarding them.

3.4.3 Control Plane

The primary function of the control plane in the SCION Internet architecture is the discovery of path segments and subsequently making them available to endpoints. The process involves three stages: path exploration, path registration, and path lookup [32].

As already pointed out in section 3.4.2, the path exploration stage encompasses a *beaconing* process. Through inter-domain beaconing (or core beaconing), core ASes learn paths to

other core ASes locally or in remote ISDs, which enables communication between different ISDs, and leads to global connectivity. Intra-domain beaconing, on the other hand, is needed so that non-core ASes learn path segments to their ISD's core. In each AS, the beacon service is responsible for regularly handling the dissemination of PCBs. As PCBs get passed from one AS to the next while traversing the network topology, they accumulate cryptographically protected and signed path data. Upon receiving a PCB, it is validated by the respective beacon server based on the signatures of the preceding ASes [1, 32].

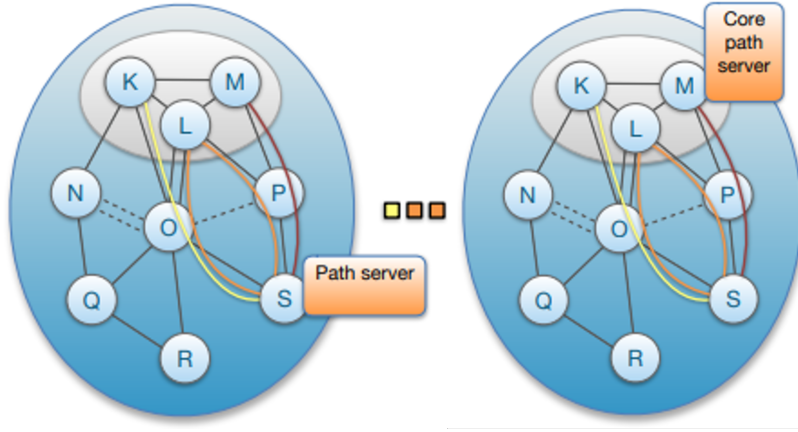


Figure 3.3: Up and down path segments. Adapted from [33], pp. 11-12.

Path registration is used to turn the information gathered in PCBs into one of three types of path segments [1]:

- **Up-path segment:** from a non-core to a core AS
- **Down-path segment:** from a core AS to a non-core AS
- **Core-path segment:** between core ASes

Path servers in an AS store up-path segments to reach the ISD core. The AS has learned these up-path segments through information they parse from PCBs received from the local beacon service. The AS is free to choose which path up-path segments should be announced to its end-hosts. Up-path segments are therefore registered at the local path service [33], as shown in the left-hand side of figure 3.3. The beacon service in an AS also selects the connections it wants to be reached through and registers those as down-path segments at the ISD core's path service. Furthermore, the local beacon service decides which PCBs they are forwarding to downstream ASes, for example, to make the corresponding path segments available to their customer. The selection process for the registration and propagation of path segments is based on a path quality metric that aims to encompass consistency, diversity, and efficiency [1, 32].

To initiate communication with a remote destination, a source host performs a path lookup to obtain the necessary path segments. Before the actual lookup, the host usually queries a SCION name service with a domain name to receive a valid SCION address for the destination, given by a triplet of the form `<ISD-ID, AS-ID, end-host address>`. The

first two elements of this address, representing identifiers for the destinations ISD and AS, are used to look up the AS-level path to the destination. The end-host address format can be specified by the targeted AS and commonly depends on the intra-AS routing protocols used. After having received a SCION address for the destination, a source host queries a local SCION path server for appropriate path segments. Up-path segments leading to the ISD core are saved in the local path server's database and can therefore be returned immediately. The server also checks its cache to see if it contains valid core- and down-path segments and if so, returns them as well. If not and the destination is in the same ISD, the local path server requests down-path segments from its ISD's core path services. Otherwise, the local path server queries its own ISD's core path services for core-path segments to the destination AS in the remote ISD. Next, it combines the received core with the known up-path segments and uses this path to query the remote ISD's path core path services for down segments to the destination. After this recursive lookup, the resulting set of up-, core-, and down-path segments are returned to the source host [1, 32].

3.4.4 Data Plane

The SCION data plane's main objective is to ensure that packet forwarding is performed according to the end-to-end paths provided by the control plane operations described in section 3.4.3. After obtaining path segments from a path lookup, there is a final step to perform by the source host before any packets can be forwarded toward their destination. The given path segments must be combined into a valid end-to-end path. A valid SCION path contains up to three path segments. There may be multiple different paths constructed from the available segments. This is a valuable property, especially in the case of disjunct paths, as it increases redundancy and enables multipath communication [1, 32].

Following the construction of possible forwarding paths, a specific instance is selected, and the path is encoded into the header of the SCION packets before transmission. This packet-carried forwarding state is cryptographically protected and contains all the necessary information for the on-path ASes to forward the packet to the next AS. This circumstance makes inter-domain routing tables superfluous, which is a major advantage over the BGP-based system. No global inter-domain routing tables and eliminating the need for longest IP-prefix matching operations on border routers means faster forwarding and the possibility of deploying more energy-conserving routers [1].

3.4.5 Security

In section 3.3.2, vulnerabilities and attack vectors in today's Internet landscape were discussed. SCION aims to achieve security by design and promises to overcome most of these concerns. This section takes a closer look at some of the security properties SCION offers.

During the beaconing process, each AS signs every PCB it forwards. This, in turn, enables the validation of a PCB by all entities, effectively removing the possibility of prefix hijackings. Another security property SCION offers is path correctness, ensuring packets are

forwarded along the intended path segments. To achieve this, the SCION packet header carries a Packet-Carried Forwarding State (PCFS), which contains cryptographically protected forwarding information. Each AS in a path segment uses a secret symmetric key to compute a message authentication code (MAC) over the forwarding information. This prevents tampering and achieves path integrity [1, 32].

SCION also offers mechanisms to defend against distributed denial of service (DDoS) attacks. End hosts can choose among different paths for communication, which increases route redundancy, enabling them to avoid congested or malicious paths and switch to alternatives. Also, enabled by SCION's path-aware networking, its multipath communication allows end hosts to use multiple paths simultaneously or switch paths dynamically in case of failures or attacks. Furthermore, SCION introduces hidden path segments that are not publicly advertised and can only be used by authorized parties. Hidden paths can provide additional protection for critical services or hosts that want to hide their location or identity, thus minimizing the risk of being targeted [1, 32].

3.4.6 Deployment

One of the key challenges for any new technology, and certainly for a new Internet architecture, is its deployment. SCION tries to facilitate adoption by providing interoperability and coexistence with existing networking technologies, thus enabling incremental deployability of SCION. An AS that wants to deploy SCION, typically an ISP, must install SCION border routers and set up path, beacon, and certificate services. No dedicated SCION hardware is needed for these components, as they run on standard servers or virtual machines. Inside an AS, SCION can be deployed as an IP overlay network, therefore, intra-domain routing does not have to be changed [1, 32].

For end-hosts, there are multiple ways of interacting with a SCION network. One option is to use an endpoint that runs SCION natively. In this configuration, a native SCION daemon can make use of path awareness and communicate with SCION-enabled applications and services. The other option is to utilize a legacy IP host, connected to a SCION proxy service, such as a SCION-IP Gateway (SIG). A SIG is a device that can translate between SCION and IP packets. It can be deployed by any entity that wants to offer or use SCION services. For example, a SIG can be deployed by a content provider that wants to deliver its content over SCION, or by an end-user that wants to access SCION-based applications [1, 32]. The SIG can be viewed as a powerful mechanism for transitioning to a SCION Internet infrastructure.

3.5 Real World Applicability of SCION

In this chapter, we discuss the real-world applicability of SCION by examining some of its current and potential deployments in various domains.

3.5.1 Industry adoption of SCION

SCION has been adopted by various domains that require high levels of security, reliability, and high availability for their network communication. This section looks at a startup commercializing SCION and expands on two domains: the financial sector as well as the research and education sector. One of the leading providers of SCION-based products is Anapaya Solutions AG, a Swiss company founded in 2017 as a spinoff from ETH Zurich. Anapaya offers a range of products that use SCION for business-oriented communications. Their solutions include a software suite for organizations, such as ISPs, to establish and manage their own SCION networks, a cloud-based service for SCION Internet access, and a SCION gateway product [40].

The Swiss financial sector is one of the environments where SCION has been deployed successfully in the form of the Secure Swiss Finance Network (SSFN). The SSFN connects Swiss banks and financial institutions, providing secure and reliable communication among its members. The network forms a SCION isolation domain, consisting of multiple core ASes operated by several Swiss ISPs, such as Swisscom and Sunrise. Members of the SSFN can also communicate to services outside the ISD, through peering links to outside ASes, including stock exchanges and other financial networks. SSFN members connect to the core ASes via encrypted VPN tunnels that run over the public Internet. Core ASes interconnect with each other through dedicated fiber links. The SSFN network provides its members with strong isolation from network attacks, high availability, and fast failover features [39].

In the field of research and education, SCION has been applied with promising results. Several academic institutions around the world have joined a global research network called SCIONLab [37], connecting their own individually run SCION ASes. The network performs inter-domain routing and enables experimentation with the properties of the SCION architecture. SCIONLab allows researchers and students to implement and test new applications and protocols based on or on top of SCION. Some of the applications that have been developed and tested in this context are video streaming, file transfer, web browsing, and blockchain technologies. Video streaming, for example, can benefit from SCION's multipath communication capabilities to achieve higher quality and lower latency. Using multiple paths concurrently can also increase throughput in file transfer. The path awareness in SCION Internet architecture and its transparency of trust roots can also benefit web browsing by enhancing user privacy and security.

3.5.2 Benefits and challenges of SCION adoption

The adoption of SCION in various domains can bring several benefits to both network operators and end users. Some of these benefits are:

Path awareness SCION grants path transparency and allows end-hosts to choose among multiple available paths for their communications based on their individual preferences [1]. This enables more control over the security and performance of their traffic

and creates the opportunity to optimize bandwidth utilization and cost [25]. An application running on an end-host can for example decide to use a path that minimizes latency, maximizes throughput, avoids congested network areas, or reduces the carbon footprint of its communications. Multipath communication features of SCION enabled by path awareness can also improve network resilience and thereby quality of service.

Failure isolation The impact of routing failures and misconfigurations is limited to a single ISD in SCION, instead of being propagated across the whole network [1]. Especially in scenarios with malicious or compromised network entities, this property can improve the resilience and availability of communication on a larger scale. For instance, if an AS in one ISD suffers from a DDoS attack, it will not affect the communication between other ASes in different ISDs, as they can use alternative paths that bypass the affected AS.

Explicit trust In SCION, end-hosts can obtain information about the trustworthiness of each path segment, thanks to the properties of the respective ISDs and their TRCs. This enables them to make informed decisions about the paths they want to use or avoid, depending on their own trust requirements [26]. Thus, any host can avoid paths that traverse untrusted ISDs or ASes or prefer paths that comply with certain regulations or policies.

However, adopting SCION also poses some challenges that need to be addressed. Some of these challenges are:

Deployment The adoption of SCION requires some changes to the existing network infrastructure, even though it may not have to be an extensive restructuring initially. Still, it would be necessary to install or upgrade at least some border routers inside each AS [1]. These border routers would ideally support both SCION and IP packets and maintain interoperability with existing routing protocols, such as BGP. Eventually, new services such as path, certificate, and beacon servers would have to be deployed as well. The additional costs and increased complexity for network operators caused by this may be a significant barrier to SCION adoption.

Scalability SCION relies on storing state in packet headers and protecting them cryptographically. This increases the size and processing overhead of packets. This may affect the scalability and efficiency of packet forwarding [27]. The architecture is designed to be highly scalable (it is in the name), but it still needs to be tested and validated under real-world conditions on a large scale to ensure that it can handle the traffic and data volumes of the modern Internet.

Adoption It may take time for end-users to understand and trust the new architecture and for service providers to develop and deploy applications and services that are compatible with SCION.

3.5.3 Comparison with other clean-slate approaches

There are other clean-slate approaches that have been proposed or implemented to address some of the limitations of the current Internet architecture.

One of the more well-known future Internet architectures is Named Data Networking (NDN) [28]. It is a data-centric approach that is proposed as a clean-slate alternative to today's host-centric IP networking. NDN is based on the idea of naming data instead of locations, using hierarchical and human-readable names. This data-centric way of communication is designed for scalability, efficiency, and security.

The Recursive InterNetwork Architecture (RINA) [29] is another clean-slate approach that tries to overcome the limitations of the current Internet. RINA builds on the principles of inter-process communication and recursion. It and tries to simplify network design, operation, and evolution.

XIA, short for eXpressive Internet Architecture [30], is a clean-slate approach motivated by the limited evolvability of the current Internet. XIA introduces a new abstraction called a principal, representing a network entity such as hosts, content, or services, that can communicate among themselves. It includes an evolvable API that allows applications to use new principals without modifying the whole network stack.

A comparison of SCION with these other clean-slate approaches reveals some similarities and also some differences. All four approaches aim to improve the security, reliability, and flexibility of the Internet, but they have different design goals and properties. While SCION focuses on providing high availability and isolation for critical networked systems, NDN focuses on providing efficient and secure data retrieval for content-centric applications. RINA aims to provide a simple and modular network architecture, and XIA wants to create a future-proof and expressive network architecture that fosters innovation.

In terms of industry adoption, the other approaches have not achieved the same level of success as SCION. NDN has been mainly deployed in testbeds and research networks [31], while XIA has been implemented as a prototype and was mostly evaluated in simulations. A prototype of RINA has also been developed, and it was demonstrated in proof-of-concept experiments, such as IRATI (Investigating RINA as an Alternative to TCP/IP) and ARCFIRE (Architectural Renovation for 5G Experimentation).

3.6 Discussion and Summary

As outlined in this report, the necessity of a clean-slate Internet architecture is justifiable, as it addresses the limitations of today's Internet, such as scalability, security, transparency, and control. In contrast to an incremental approach, the clean slate architecture has the advantage of not being constrained by preexisting designs. Replacing the current Internet architecture with a clean-slate one, such as SCION, is a challenging but feasible goal. Several factors may influence the feasibility of such a transition.

The architecture's design takes into account the compatibility of SCION with the existing Internet infrastructure and its protocols. Interoperability of SCION with IP is ensured by the SCION-IP Gateway (SIG) service, which allows users to access SCION services from IP networks and vice versa. The SIG service handles the encapsulation and decapsulation of the different network packets, translating them into the needed format, as well as performing some routing and forwarding tasks [1]. This makes the incremental adoption of SCION easier and thereby increases the feasibility of the clean-slate approach.

SCION adoption also offers many technical and economic incentives for different stakeholders. ISPs can create new SCION-based services that generate revenue streams, and add value for customers such as higher availability, without the cost of dedicated leased lines. SCION also allows for finer-grained control in traffic engineering through expressive PCB propagation policies. Businesses can ensure their compliance with privacy laws, profiting from SCION's path-control capabilities. The performance and availability advantages advertised by SCION, combined with its increased security assurances, make it also attractive for end-users [1].

Another aspect to point out is the different governance models and policies that are supported by the SCION architecture. The concept of ISDs that define their own TRCs enables them to achieve autonomy and sovereignty over a network domain while keeping the ability to interconnect with other ISDs for global connectivity.

Some of the challenges associated with adopting SCION were highlighted, such as deployability, scalability, and convincing stakeholders to adopt the new technology. Even though this report discussed several successful projects implementing SCION, the production deployments that were seen so far are mostly focused on business-to-business communication. Therefore, it remains to be seen, if the SCION Internet architecture is suitable to replace the Internet as we know it today in its entirety.

However, this report discussed many interesting and useful properties and use cases of the SCION Internet architecture, that make it a promising candidate among the proposals for a clean-slate Internet. SCION could improve the security, reliability, and efficiency of end-to-end communication and guarantee higher network availability. It also brings new functionalities such as path-aware and multipath communication, creating opportunities for innovation. Furthermore, the approach can coexist with the current IP architecture, allowing a gradual transition to SCION.

Therefore, it can be argued that replacing the current Internet architecture with SCION is feasible but challenging. Nevertheless, SCION has the potential to become a viable alternative to the current Internet architecture.

Bibliography

- [1] Adrian Perrig, Pawel Szalachowski, Raphael M. Reischuk, Laurant Chuat: *SCION: A secure Internet architecture*, Springer, Cham, October 2017, <https://doi.org/10.1007/978-3-319-67080-5>.
- [2] Yakov Rekhter, Susan Hares, Tony Li: *A Border Gateway Protocol 4 (BGP-4)*, RFC 4271, January 2006, <https://www.rfc-editor.org/info/rfc4271>.
- [3] Kevin Butler, Toni R. Farley, Patrick McDaniel, Jennifer Rexford: *A Survey of BGP Security Issues and Solutions*, Proceedings of the IEEE, Vol. 98, No. 1, January 2010, pp. 100-122, <https://doi.org/10.1109/JPROC.2009.2034031>.
- [4] Andrew S. Tanenbaum, David Wetherall, Nick Feamster: *Computer Networks*, Pearson Education, Limited, Harlow, England, Sixth edition, July 2021.
- [5] Ramesh Govindan, Anoop Reddy: *An Analysis of Internet Inter-Domain Topology and Route Stability*, Proceedings of INFOCOM, IEEE, Kobe, Japan, Vol. 2, April 1997, pp. 850-857
- [6] David A. Maltz, Geoffrey Xie, Jibin Zhan, Hui Zhang, Gisli Hjattmysson, Albert Greenberg: *Routing design in operational networks: A look from the inside*, ACM SIGCOMM Computer Communication Review, Vol. 34, No. 4, October 2004, pp. 27-40, <https://dl.acm.org/doi/10.1145/1030194.1015472>.
- [7] Jon Postel: *Internet Protocol*, RFC 791, September 1981, <https://www.rfc-editor.org/info/rfc791>.
- [8] Kai Bu, Avery Laird, Yutian Yang, Lingfeng Cheng, Jiaqing Luo, Yingjiu Li, Kui Ren: *Unveiling the mystery of internet packet forwarding: A survey of network path validation*, ACM Computing Surveys (CSUR), ACM New York, USA, Vol. 53, No. 5, September 2021, pp. 1-34, <https://doi.org/10.1145/3409796>
- [9] John Moy: *OSPF Version 2*, RFC 2328, April 1998, <https://www.rfc-editor.org/info/rfc2328>.
- [10] David R. Oran: *OSI IS-IS Intra-domain Routing Protocol*, RFC 1142, February 1990, <https://www.rfc-editor.org/info/rfc1142>.
- [11] Geoff Huston: *BGP in 2022 – the routing table* APNIC Blog, January, 2023, <https://blog.apnic.net/2023/01/06/bgp-in-2022-the-routing-table/>, last visit: April, 2023.

- [12] Geoff Huston: *BGP in 2022 – BGP updates* APNIC Blog, January, 2023, <https://blog.apnic.net/2023/01/11/bgp-in-2022-bgp-updates/>, last visit: April, 2023.
- [13] Banit Agrawal, Timothy Sherwood: *Modeling TCAM power for next generation network devices*, IEEE International Symposium on Performance Analysis of Systems and Software, Austin, TX, USA, March 2006, pp. 120-129, <https://doi.org/10.1109/ISPASS.2006.1620796>.
- [14] Ricardo Bennesby da Silva, Edjard Souza Mota: *A survey on approaches to reduce BGP interdomain routing convergence delay on the Internet*, IEEE Communications Surveys & Tutorials, Vol. 19, No. 4, June 2017, pp. 2949-2984, <https://doi.org/10.1109/COMST.2017.2722380>.
- [15] Marcelo Yannuzzi, Xavier Masip-Bruin, Olivier Bonaventure: *Open issues in inter-domain routing: a survey*, IEEE network, Vol. 19, No. 6, November 2005, pp. 49-56, <https://doi.org/10.1109/MNET.2005.1541721>.
- [16] Asya Mitseva, Andriy Panchenko, Thomas Engel: *The state of affairs in BGP security: A survey of attacks and defenses*, Computer Communications, Vol. 124, June 2018, pp. 45-60, <https://doi.org/10.1016/j.comcom.2018.04.013>
- [17] Spencer Dawkins: *Path Aware Networking: Obstacles to Deployment (A Bestiary of Roads Not Taken)*, RFC 9049, June 2021, <https://www.rfc-editor.org/info/rfc9049>.
- [18] *Path Aware Networking RG (Panrg)*, IETF, <https://datatracker.ietf.org/rg/panrg/about/>, last visit: April, 2023.
- [19] Brian Trammell, Jean-Pierre Smith, Adrian Perrig: *Adding path awareness to the Internet architecture*, IEEE Internet Computing, Vol. 22, No. 2, April 2018, pp. 96-102, <https://doi.org/10.1109/MIC.2018.022021673>.
- [20] Charlie Kaufman, Radia Perlman, Mike Speciner: *Network security: private communication in a public world*, Addison-Wesley, Boston, USA, Third edition, September 2022.
- [21] Spencer Dawkins: *The Transport Layer Security (TLS) Protocol Version 1.3*, RFC 8446, August 2018, <https://www.rfc-editor.org/info/rfc8446>.
- [22] Xin Hu, Z. Morley Mao: *Accurate real-time identification of IP prefix hijacking*, IEEE Symposium on Security and Privacy, IEEE, Berkeley, CA, USA, June 2007, <https://doi.org/10.1109/SP.2007.7>.
- [23] Robert McMillan: *YouTube Outage Underscores Big Internet Problem*, InfoWorld, IDG News Service, 26 Feb. 2008, <https://www.infoworld.com/article/2648947/youtube-outage-underscores-big-internet-problem.html>, last visit: March, 2023.
- [24] M. Lepinski, Ed., K. Sriram, Ed.: *BGPsec Protocol Specification*, RFC 8205, September 2017, <https://www.rfc-editor.org/info/rfc8205>.

- [25] Simon Scherrer, Markus Legner, Adrian Perrig, Stefan Schmid: *An Axiomatic Perspective on the Performance Effects of End-Host Path Selection*, ACM SIGMETRICS Performance Evaluation Review, Vol. 49, No. 3, March 2022, pp. 16-17, <https://doi.org/10.1145/3529113.3529118>.
- [26] Xin Zhang, Hsu-Chun Hsiao, Geoffrey Hasker, Haowen Chan, Adrian Perrig, David G. Andersen: *SCION: Scalability, Control, and Isolation on Next-Generation Networks*, IEEE Symposium on Security and Privacy, IEEE, Oakland, CA, USA, May 2011, pp. 212-227, <https://doi.org/10.1109/SP.2011.45>.
- [27] David Barrera, Raphael M. Reischuk, Pawel Szalachowski, Adrian Perrig: *SCION Five Years Later: Revisiting Scalability, Control, and Isolation on Next-Generation Networks*, August 2015, <https://doi.org/10.48550/arXiv.1508.01651>
- [28] Lixia Zhang, Alexander Afanasyev, Jeffrey Burke, Van Jacobson, kc claffy, Patrick Crowley, Christos Papadopoulos, Lan Wang, Beichuan Zhang: *Named data networking*, SIGCOMM Comput. Commun. Rev., Vol. 44, No. 3, July 2014, pp. 66-73 <https://doi.org/10.1145/2656877.2656887>.
- [29] *What Is RINA?* | RINA, Boston University Computer Science: RINA, <https://csr.bu.edu/rina/about.html>, last visit: April, 2023.
- [30] Ashok Anand, Fahad Dogar, Dongsu Han, Boyan Li, Hyeontaek Lim, Michel Machado, Wenfei Wu, Aditya Akella, David G. Andersen, John W. Byers, Srinivasan Seshan, Peter Steenkiste: *XIA: an architecture for an evolvable and trustworthy internet*, In Proceedings of the 10th ACM Workshop on Hot Topics in Networks (HotNets-X), Association for Computing Machinery, New York, NY, USA, Article 2, November 2011, pp. 1-6, <https://doi.org/10.1145/2070562.2070564>.
- [31] *NDN Testbed*, Named Data Networking (NDN), May 2022, <https://named-data.net/ndn-testbed/>, last visit: April, 2023.
- [32] Corine de Kater, Nicola Rustignoli, Adrian Perrig: *SCION Overview*, Internet-Draft, draft-dekater-panrg-scion-overview-03, Internet Engineering Task Force, March 2023, <https://datatracker.ietf.org/doc/draft-dekater-panrg-scion-overview/03/>, last visit: April, 2023.
- [33] Adrian Perrig: *SCION: Architecture Overview*, Network Security Group, ETH Zürich, scion-architecture.net/slides/SL-architecture-overview.pdf, last visit: April, 2023.
- [34] Constantine Dovrolis: *What would Darwin think about clean-slate architectures?*, ACM SIGCOMM Computer Communication Review, Vol. 38, No. 1, January 2008, pp. 29-34, <https://doi.org/10.1145/1341431.1341436>.
- [35] David D. Clark: *The design philosophy of the darpa internet protocols*, SIGCOMM Comput. Commun. Rev., Vol. 18, No. 4, August 1988, pp. 106-114, <https://doi.org/10.1145/52324.52336>
- [36] Radio Perlman: *An overview of PKI trust models*, IEEE Network, Vol. 13, No. 6, November-December 1999, pp. 38-43, <https://doi.org/10.1109/65.806987>.

- [37] *Scionlab* SCIONLab, www.scionlab.org/, last visit: April, 2023.
- [38] Steven M. Bellovin: *A look back at" security problems in the tcp/ip protocol suite*, 20th Annual Computer Security Applications Conference, IEEE, Tucson, AZ, USA, December 2004, pp.229-249, <https://doi.org/10.1109/CSAC.2004.3>
- [39] *Secure Swiss Finance Network (SSFN)*, Secure Swiss Finance Network (SSFN) | SIX, SIX Group, www.six-group.com/en/products-services/banking-services/ssfn.html, last visited: June, 2023.
- [40] *Anapaya - The SCION Internet*, Anapaya, www.anapaya.net/, last visited: June, 2023.

Chapter 4

ML Models for Textual Data Anonymization in NLP

Nicolas Kohler and Ramize Abdili

Since the introduction of the GDPR in 2018, companies and researchers alike are spending more time pondering about which data they use and how. However, anonymizing data for market predictions or research can be a very time-consuming and manual task, especially when considering unstructured data, like text. This becomes even more important in industries where privacy is key, like financial institutions and healthcare. Hence, this paper discusses the existing techniques available to automate this otherwise manual and time-consuming task. To do this, we first elaborate on various privacy models such as K-Anonymity, L-Diversity, and Differential Privacy. Having laid the theoretical foundation, we then discuss how Machine Learning based Natural Language Processing, especially with regards to Named Entity Recognition, can be used to identify sensitive information in text, which then can be anonymized in a second step via a tool. Lastly, we will also discuss how Deep Learning and Transformers, especially Chat GPT, could be used to aid in anonymization tasks, as they provide an end-to-end solution that no longer requires applying a specific anonymization tool to previously identified entities.

Contents

4.1	Introduction	73
4.2	Data Anonymization Techniques	73
4.2.1	Anonymizing Direct Identifiers	74
4.2.2	Anonymizing Quasi Identifiers	76
4.3	Privacy Models for Data Anonymization Techniques	79
4.3.1	K-Anonymity	79
4.3.2	Attacks of K-Anonymity	80
4.3.3	L-Diversity	81
4.3.4	Differential Privacy	81
4.4	Natural Language Processing for Textual Data Anonymization	83
4.4.1	Overview of Named Entity Recognition Approaches	83
4.4.2	Using Identified Named Entities For Anonymization	84
4.5	Machine Learning based NLP for Textual Data Anonymization	85
4.5.1	SVM	85
4.5.2	CRF	86
4.6	Deep Learning based NLP for textual Data Anonymization	86
4.6.1	Transformer Architecture	87
4.6.2	Training Transformers	89
4.6.3	Generative Pretrained Transformer (GPT)	89
4.6.4	ChatGPT for Data Anonymization	90
4.7	Discussion and Conclusion	92

4.1 Introduction

At least since the introduction of the GDPR data privacy protection and thus anonymization became a more pressing issue [19]. With a law that applies to all data of european citizens, no matter where it is processed, even companies outside of the european union are forced to comply [20]. One way to achieve compliance is by anonymizing data, as the GDPR does not apply to neither anonymized nor synthetic data [39]. Furthermore, while the law demands compliance with regards to data privacy, it might also be in the business interests to anonymize or de-identify data to allow for easier data sharing with third parties [1]. Additionally, maintaining anonymized data also is beneficial when considering security. As, given a data breach, the attackers might not be able to access confidential information or reveal the identity of the person appearing in the data [21].

The above only highlights three out of many reasons why an entity would like to, or has to, perform anonymization. However, to perform anonymization fast and at scale, one can not rely on manual approaches to do data anonymization. That is especially true when dealing with raw textual data, where the first step is to identify the entities that have to be anonymized. Thankfully, one can apply automatic or semi-automatic techniques by using NLP [22]. With NLP sensitive entities can be automatically classified and extracted from text, allowing the anonymization in a second step. More recent advancements in Deep Learning and Transformers can integrate these two steps into one.

However, before diving into Machine Learning and Transformer based data anonymization we first have to recap on anonymization techniques and privacy models. Such a recap will be provided in Sections 4.2 and 4.3. Following this theoretical foundation we will provide an overview of how Natural Language Processing can be used for Textual Data Anonymization in Section 4.4. This overview then allows us to dive deeper into Machine Learning based NLP in Section 4.5 and Deep Learning based NLP in Section 4.6, before closing with a discussion and conclusion in Section 4.7.

4.2 Data Anonymization Techniques

Data anonymization techniques are necessary to protect sensitive data in a given dataset and reduce the risk of identifying individuals. The primary goal is to increase data privacy while still allowing meaningful analysis of the data. These techniques can be categorized into two main groups: those that anonymize direct identifiers and those that anonymize quasi-identifiers. Direct identifiers are specific pieces of information within data records that have the ability to directly identify an individual, while quasi-identifiers refer to data entries that, by themselves, do not directly identify an individual [1]. However, when quasi-identifiers are combined or linked with other information, they have the potential to enable the identification of the individual [1]. While we will focus on direct and quasi-identifiers, it is worth noting that some sources also differentiate between confidential and non-confidential attributes [40]. Confidential attributes refer to private and sensitive information about an individual that is not commonly known, such as personal medical details [40]. Non-confidential attributes, on the other hand, encompass more general

information that doesn't directly identify a person but can still be used in combination with other data to gain insights or draw connections [40]. However, in the following subsections, we will focus on the anonymization of direct and quasi-identifiers.

4.2.1 Anonymizing Direct Identifiers

Direct identifiers are data records that can be used to directly identify an individual [1]. Figure 4.1 shows an example data table with columns for name, address, birthday, ZIP code, weight, and diagnosis. In this case, the direct identifiers are the "Name" and "Address" columns. To achieve anonymization, direct identifiers must undergo a transformation to ensure that no individual in the dataset can be identified. Common techniques for anonymizing direct identifiers include removing them entirely, replacing them, data masking, and data pseudoanonymization [1].

Direct Identifiers								
Name	Address	Birthday	ZIP	Sex	Weight	Diagnosis	...	...

Figure 4.1: Example of a data table with its direct identifiers in red [1]

4.2.1.1 Removing

The simplest way to anonymize data is by removing all direct identifiers from the database, as this eliminates the risk of individuals being directly identified [1]. This approach is widely used and is often effective in protecting privacy [1].

4.2.1.2 Replacing

Another approach for anonymizing direct identifiers is by replacing them with less identifying information. This involves substituting direct identifiers with generic or categorical names [1]. For instance, data in the "Name" column could be replaced with the category name "PERSON NAME," while data in the "Address" column could be replaced with the text "123 ANY ROAD, ANY TOWN, ANY COUNTRY". Figure 4.2 illustrates this replacement approach. While this technique can be effective in protecting privacy, it can also lead to a loss of detail and accuracy in the data [1].

4.2.1.3 Data Masking

Data masking is another approach to anonymize direct identifiers. In this technique, direct identifiers are replaced with a repeating symbol, such as "XXX" or "***", with the

Name	Address	...
John Smith	123 Main St, New York, USA	
Sarah Johnson	456 Maple Ave, San Diego, USA	
David Lee	789 Oak St, Washington, USA	

Name	Address	...
PERSON NAME	123 ANY ROAD, ANY TOWN, ANY COUNTRY	
PERSON NAME	456 ANY ROAD, ANY TOWN, ANY COUNTRY	
PERSON NAME	789 ANY ROAD, ANY TOWN, ANY COUNTRY	

Figure 4.2: Replacing the direct identifiers; adapted from [1]

length of the symbol matching the length of the original data. For example as illustrated in Figure 4.3, the name "John" can be masked with "XXXX," while the name "Sarah" could be masked with "XXXXX," or alternatively with "*****" and "*****" and so on. While data masking can be effective in protecting privacy, it may also reduce the value and accuracy of the data [1].

Name	Address	...
John Smith	123 Main St, New York, USA	
Sarah Johnson	456 Maple Ave, San Diego, USA	
David Lee	789 Oak St, Washington, USA	

Name	Address	...
XXXX XXXX	XXXX XXXX, XXXXXXXX, XXX	
XXXXX XXXXX	XXXX XXXX, XXXXXXXX, XXX	
XXXX XXX	XXXX XXX, XXXXXXXX, XXX	

Figure 4.3: Masking the direct identifiers

4.2.1.4 Data Pseudoanonymization

The last approach to anonymize direct identifiers is data pseudoanonymization. It replaces the direct identifiers with pseudonyms that are generated randomly or through an algorithm. In this approach, direct identifiers that are the same are replaced with the same pseudonym value. An example is shown in Figure 4.4. However, data pseudoanonymization comes with some risks. If the parameters of the algorithm, that generated the pseudonyms are discovered, the transformation can be reversed, revealing the original values of the direct identifiers. This possibility is known as a brute-force attack [1]. Additionally, even if direct identifiers are replaced with pseudonyms that cannot be reversed, an individual can still be re-identified by linking multiple datasets in a so-called linkage attack [1]. Therefore, data pseudoanonymization may not be highly secure for anonymizing direct identifiers, since there is still a risk of re-identification [1, 2].

Name	Address	...
John Smith	123 Main St, New York, USA	
Sarah Johnson	456 Maple Ave, San Diego, USA	
David Lee	789 Oak St, Washington, USA	

Name	Address	...
Delta Echo	123 Cedar Rd, Anytown, Kylonia	
Tango Golf	456 Pine St, Anytown, Novaria	
Charlie Hotel	789 Oak Ave, Anytown, Aquilonia	

Figure 4.4: Pseudoanonymization of the direct identifiers

4.2.2 Anonymizing Quasi Identifiers

After enumerating the methods for anonymizing direct identifiers, this section will explore techniques for anonymizing quasi-identifiers. Quasi-identifiers are data records in a dataset that do not directly identify an individual, but when linked with other information, can lead to identification [1]. In the table shown in Figure 4.5, the columns "Birthday," "ZIP," and "Sex" are examples of quasi-identifiers, highlighted in violet. The following section will discuss how to anonymize such quasi-identifiers.

Direct Identifiers		Quasi-Identifiers						
Name	Address	Birthday	ZIP	Sex	Weight	Diagnosis	...	...

Figure 4.5: Example of a data table with its quasi-identifiers in violet [1]

4.2.2.1 Suppression

One common technique for anonymizing quasi-identifiers is called suppression. This involves removing or suppressing quasi-identifiers from a database. While this approach can be effective in protecting the individuals' privacy, it also has drawbacks. Because quasi-identifiers often provide important information about the characteristics of individuals in a dataset, suppressing them can reduce the overall usefulness of the data [1]. As a result, alternative approaches to anonymization may be more appropriate depending on the specific context and goals of the analysis [1].

4.2.2.2 Data Perturbation

An effective technique for anonymizing quasi-identifiers is data perturbation, which involves modifying the quasi-identifiers by adding noise or altering their values [1, 2]. For example, a common approach is to shift all birthday dates in a dataset by a fixed number of days, or to add or subtract a constant value from all age data [1, 2]. As illustrated in Figure 4.6, data perturbation can be used to increase the age records by a constant value, substitute the birthday records with values that are a certain number of days earlier or later, add a constant value to the ZIP codes, while leaving the sex data unchanged.

...	Age	Birthday	ZIP	Sex	...
	33	10/03/1990	10001	male	
	37	25/06/1985	10002	female	
	27	15/11/1995	10003	male	



...	Age	Birthday	ZIP	Sex	...
	35	15/02/1990	10005	male	
	39	02/06/1985	10006	female	
	29	23/10/1995	10007	male	

Figure 4.6: Data Perturbation

Data perturbation is often used for numeric quasi-identifiers [2]. However, care must be taken to balance the level of perturbation to maintain the accuracy and usefulness of the data while still protecting individuals' privacy [1, 2].

4.2.2.3 Data Swapping

The data swapping technique involves randomly shuffling the quasi-identifiers in a dataset, so that they can be exchanged between data records, making it impossible to infer any meaningful information from them [1, 2]. As illustrated in Figure 4.7, this technique shuffles the values of quasi-identifiers, such as birthdays, between records in a dataset. For instance, if the first individual in the original table has an age of 33, a birthday of 10/03/1990, a zip code of 10001, and a sex of Male, we can swap their values with one of the other individuals in the table to create a new row in the swapped data table. This process is repeated for each individual in the original table, resulting in a new table with the same number of rows and columns, but with the values shuffled between rows. This approach is aimed at preventing re-identification while preserving data utility [1, 2].



...	Age	Birthday	ZIP	Sex	...
	33	10/03/1990	10001	male	
	37	25/06/1985	10002	female	
	27	15/11/1995	10003	male	

...	Age	Birthday	ZIP	Sex	...
	33	15/11/1995	10003	male	
	37	10/03/1990	10001	male	
	27	25/06/1985	10002	female	

Figure 4.7: Data Swapping

4.2.2.4 Sub Sampling

Another technique is sub sampling, which involves randomly selecting a subset of the original dataset to keep, while discarding the rest [1]. This helps to protect sensitive data from being identified since only a subset of the original data is made public. Sub sampling is illustrated in Figure 4.8, where a random subset of rows is retained, while the others are discarded. However, the values in each column still preserve the general characteristics of the data. For instance, in the anonymized table, the row that corresponds to the individual who was originally in the second row of the original table (i.e., the person who was 37 years old, had a birthday of 25/06/1985, lived in zip code 10002, and was female) is not included. By reducing the amount of data that is published, sub sampling can reduce the risk of re-identification of individuals in the dataset [1, 2].



...	Age	Birthday	ZIP	Sex	...
	33	10/03/1990	10001	male	
	37	25/06/1985	10002	female	
	27	15/11/1995	10003	male	

...	Age	Birthday	ZIP	Sex	...
	33	10/03/1990	10001	male	
	27	15/11/1995	10003	male	

Figure 4.8: Sub Sampling

4.2.2.5 Synthetic Data

The synthetic data technique is an alternative method for anonymizing a database, which involves generating a new dataset that resembles the original dataset [2]. The generated dataset contains fully or partially synthetic data that does not correspond to actual individuals in the original dataset, thus preventing identification [2]. This approach aims to preserve the statistical properties of the original dataset, while ensuring that no individual's identity is compromised [2].

As shown in Figure 4.9, a synthetic dataset is generated that closely resembles the original dataset. The new values are statistically similar to the original values but do not reveal any identifiable information about the individuals in the dataset. For instance, the distribution of ages in the anonymized dataset is similar to the distribution in the original dataset, but the exact ages are not the same. By applying this technique, we can create a new table that preserves the general characteristics of the original dataset, while ensuring that no individual's identity is disclosed.



...	Age	Birthday	ZIP	Sex	...
	33	10/03/1990	10001	male	
	37	25/06/1985	10002	female	
	27	15/11/1995	10003	male	

...	Age	Birthday	ZIP	Sex	...
	34	10/02/1991	10004	female	
	36	20/06/1985	10005	unknown	
	29	20/11/1995	10002	male	

Figure 4.9: Synthetic Data

4.2.2.6 Data Generalization

Data generalization is a technique used to anonymize quasi-identifiers by replacing some of their values with a given range or more generalized values [1]. This method helps to preserve the usefulness of the data while still protecting sensitive information [1]. For example, zip codes can be replaced with a more general value, like the first three digits, to hide specific location information, while preserving the general geographic region [1]. Similarly, age records can be generalized by grouping them into categories, such as age ranges of 10 years, to represent a group of people with similar characteristics [1]. Figure 4.10 illustrates this approach where the original age values are generalized to a range representing a group of people with similar characteristics. In addition, the specific zip code values are replaced with a more general value of 10000 to hide the exact location information, while still preserving the general geographic region. For the birthday column, the date can be generalized to the month and year or even just the year. However, to further protect privacy, the birth year can be generalized to the decade (i.e., replacing the last digit with "X"), making it harder to identify individuals based on their exact birth-date. This technique helps in preventing the re-identification of individuals while still maintaining the general characteristics of the data [1].

...	Age	Birthday	ZIP	Sex	...
	33	10/03/1990	10001	male	
	37	25/06/1985	10002	female	
	27	15/11/1995	10003	male	

...	Age	Birthday	ZIP	Sex	...
	30-39	03/199X	10000	male	
	30-39	06/198X	10000	female	
	20-29	11/199X	10000	male	

Figure 4.10: Data Generalization

4.3 Privacy Models for Data Anonymization Techniques

In this section, we will discuss three main privacy models for data anonymization techniques: K-Anonymity, L-Diversity, and Differential Privacy. While there are other privacy models available, such as k-Map, Average Risk, Population Uniqueness, Sample Uniqueness, t-Closeness, Delta-Disclosure Privacy, Beta-Likeness, Delta-Presence, and Profitability, we will focus on the three main models in this discussion [9].

4.3.1 K-Anonymity

The first privacy model of data anonymization techniques is known as K-Anonymity, which is designed to protect the privacy of individuals in a dataset. The primary goal of K-Anonymity is to prevent the linking of a dataset to re-identify individuals, which is a critical issue in privacy protection [7]. To achieve this goal, K-Anonymity requires the consideration of certain quasi-identifiers in the dataset, which are attributes that can be used to distinguish individuals from others [7].

K-Anonymity ensures that each person's data in a dataset differs from at least k other individuals' data in the same dataset regarding the quasi-identifiers [9]. In other words, a dataset is said to be k -anonymous if, for every combination of quasi-identifiers, there are at least k -matching records [1]. The larger the k value is, the higher the level of privacy protection [4]. K-Anonymity achieves this by ensuring that no one in the table can be distinguished from at least $k-1$ other individuals within the same table [4]. This prevents anyone from re-identifying an individual's personal data when it is released publicly. For example, data from a hospital that includes information about medications is considered sensitive because it can reveal a person's medical condition [3].

To illustrate the concept of K-Anonymity, let's consider a table containing the information on ZIP code, age, nationality, and health condition of twelve people, as shown in Figure 4.11a. By applying anonymization techniques, we can achieve $k=4$ anonymity, as shown in Figure 4.11b. In this case, each row in the table with the same quasi-identifiers "Zip Code", "Age", and "Nationality" appears at least four times with a different medical condition, ensuring privacy protection [5].

While K-Anonymity provides a good level of privacy protection, it has some limitations and does not guarantee complete privacy protection. There are several attacks against this property that could re-identify individuals in the dataset [3]. Therefore, K-Anonymity is often used in combination with other privacy techniques to enhance privacy protection [3].

It is also worth noting, that K-Anonymity is not sufficient to counterattack the attacks because it leaks information in groups [4]. Even if the quasi-identifiers are protected with k-anonymity, the information leakage in groups may reveal sensitive information about individuals. Therefore, it is important to consider other privacy techniques in combination with K-Anonymity to provide a higher level of privacy protection [4].

	Non-Sensitive			Sensitive
	Zip Code	Age	Nationality	Condition
1	13053	28	Russian	Heart Disease
2	13068	29	American	Heart Disease
3	13068	21	Japanese	Viral Infection
4	13053	23	American	Viral Infection
5	14853	50	Indian	Cancer
6	14853	55	Russian	Heart Disease
7	14850	47	American	Viral Infection
8	14850	49	American	Viral Infection
9	13053	31	American	Cancer
10	13053	37	Indian	Cancer
11	13068	36	Japanese	Cancer
12	13068	35	American	Cancer

	Non-Sensitive			Sensitive
	Zip Code	Age	Nationality	Condition
1	130**	< 30	*	Heart Disease
2	130**	< 30	*	Heart Disease
3	130**	< 30	*	Viral Infection
4	130**	< 30	*	Viral Infection
5	1485*	≥ 40	*	Cancer
6	1485*	≥ 40	*	Heart Disease
7	1485*	≥ 40	*	Viral Infection
8	1485*	≥ 40	*	Viral Infection
9	130**	3*	*	Cancer
10	130**	3*	*	Cancer
11	130**	3*	*	Cancer
12	130**	3*	*	Cancer

(a) No K-Anonymity applied [5]

(b) K-Anonymity with K=4 [5]

Figure 4.11: K-Anonymity Example

4.3.2 Attacks of K-Anonymity

As the digital world evolves, attackers become more sophisticated in their methods to re-identify individuals. As a result, K-Anonymity has become vulnerable to various attacks [4]. Two of the most well-known attacks are the Homogeneity attack and the Background Knowledge attack [4].

The Homogeneity attack is a well-known privacy breach in K-anonymity, which highlights the vulnerability of the approach to information leakage [4]. To illustrate this attack, let's consider two individuals, Sarah and John, who are close friends. One day, John is admitted to the hospital, and Sarah becomes curious to know about his health condition. If Sarah has access to a k-anonymous table shown in Figure 4.11b and knows that John is 31 years old, an American, and lives in ZIP code 13053, she can easily deduce which records in the table could correspond to John. Specifically, John's health status could only be one of four possibilities, all of which have the same health condition - cancer - and are clustered together in the k-anonymous table (records number 9, 10, 11, and 12). As a result, Sarah can quickly re-identify John and deduce his sensitive medical information - that he has cancer [5]. This example demonstrates that K-anonymity can create groups in a dataset that leak information, and sensitive data are no longer anonymous to everyone.

The Background Knowledge Attack is another method used to exploit the vulnerabilities of the k-anonymity privacy model. In this attack scenario, let us consider two individuals named Sarah and May. May is a 21-year-old Japanese female who lives in ZIP code 13068. Once again, Sarah has access to the table from Figure 4.11b and she can conclude that May must be one of the records numbered 1, 2, 3, or 4 based on the quasi-identifier attributes such as age, nationality, and ZIP code. From this information, Sarah can further

infer that May has either a viral infection or heart disease as all the records belonging to the aforementioned group share these two conditions. However, Sarah can also leverage her background knowledge to deduce that Japanese individuals have a significantly lower likelihood of developing heart disease. Therefore, Sarah can safely assume that May must have a viral infection rather than heart disease. The above examples followed the explanation in the paper "l-diversity: Privacy beyond k-anonymity" [5].

It is worth noting, that the sensitive value for a set of k records can be predicted when all the values within that set are identical. Therefore, the lack of diversity in the sensitive attribute values can easily enable attackers to re-identify individuals in k -anonymous tables [4]. In addition to the Homogeneity and Background Knowledge attacks, there are many other attacks that can exploit K-anonymity's vulnerabilities, such as the Minimality attack, Unsorted Matching attack, Complementary Release attack, and Temporal attack [4]. Therefore, to protect sensitive data in a dataset, it is necessary to take these attacks into account and employ additional privacy models to supplement k -anonymity [4].

4.3.3 L-Diversity

While K-Anonymity falls short in providing sufficient privacy protection, L-Diversity is a privacy model that addresses the limitations of K-Anonymity by requiring each block of data to have at least l well-represented values of the sensitive attribute. This makes it more difficult for an attacker to re-identify individuals [5]. In other words, each group or equivalence class in the dataset, must contain a minimum number of unique sensitive attribute values in order to be considered diverse [5]. If each block in the dataset satisfies this principle, the entire dataset is considered L-Diverse [5]. The benefit of this model is enhanced privacy protection against attacks on sensitive data while preserving data utility [5, 4].

To demonstrate how L-Diversity can prevent attacks on the dataset, we can transform the original table from Figure 4.11b into a 3-diverse table, as shown in Figure 4.12. By comparing the original and the 3-diverse table, it becomes clear that the latter provides better privacy protection. Even if Sarah knows the age, nationality, gender, and ZIP code of an individual, it becomes challenging for her to determine their health condition [5]. With more than three options to choose from, Sarah cannot directly conclude the health condition of an individual based on non-sensitive values alone even if she has background knowledge [5]. Therefore by using L-Diversity, we can ensure that our datasets are more robust against privacy attacks and can better protect the sensitive information of individuals [5].

4.3.4 Differential Privacy

Differential Privacy is a powerful privacy model that aims to protect the privacy of individuals in a dataset by limiting the increasing re-identification risk when new individuals

	Non-Sensitive			Sensitive
	Zip Code	Age	Nationality	Condition
1	1305*	≤ 40	*	Heart Disease
4	1305*	≤ 40	*	Viral Infection
9	1305*	≤ 40	*	Cancer
10	1305*	≤ 40	*	Cancer
5	1485*	> 40	*	Cancer
6	1485*	> 40	*	Heart Disease
7	1485*	> 40	*	Viral Infection
8	1485*	> 40	*	Viral Infection
2	1306*	≤ 40	*	Heart Disease
3	1306*	≤ 40	*	Viral Infection
11	1306*	≤ 40	*	Cancer
12	1306*	≤ 40	*	Cancer

Figure 4.12: L-diversity with L=3 [5]

are added [6]. To achieve this, random noise is introduced to the dataset, making it difficult for attackers to obtain information about specific individuals and providing plausible deniability for all members of the dataset [1, 6, 7].

The amount of privacy protection provided by Differential Privacy is measured by the parameter epsilon (ϵ), using a randomized algorithm [1, 7]. For an algorithm to satisfy the (ϵ)-differential privacy property, it must ensure that the presence or absence of a single individual in two different datasets does not significantly affect the output of the analysis [6, 7]. This is achieved by adding carefully balanced random noise to the dataset [6, 7]. The trade-off between privacy protection and data accuracy is optimized by adjusting the amount of noise added. Specifically, the probability of an individual being in one dataset divided by the probability of the individual being in another dataset should be no greater than the parameter ϵ [6, 7]. As the parameter size decreases, the amount of noise in the dataset increases, making it increasingly challenging to differentiate between individuals in the dataset [1]. This ensures that the privacy of individuals in the dataset is protected [1].

Differential Privacy provides strong privacy protection for each individual in the dataset, even if one individual is added or removed from the dataset, as the added noise masks the individual and ensures that the output of the database remains the same [1]. Although Differential Privacy is non-truthful because it relies on adding noise to the dataset [8], which can reduce the accuracy of the analysis [1], it is a powerful privacy model that protects datasets from identity disclosure [1]. Overall, Differential Privacy ensures that sensitive data can be analyzed, while protecting the privacy of individuals, even if an attacker has some knowledge about the dataset and the individuals in it [9].

4.4 Natural Language Processing for Textual Data Anonymization

After having discussed anonymization techniques and privacy models, we can now move on to Natural Language Processing for anonymization. Let's assume we would like to anonymize the following sentence: "Peter Parker is suffering from a Spider bite". One way to anonymize the data is to read the text and manually identify the sensitive information. In our example, we can see that "Peter Parker" is a name that is associated with a health condition. One could anonymize this sentence by replacing the name "Peter Parker" with "X X", therefore the condition spider bite can no longer be linked to Peter Parker. This is pretty straightforward and easily doable for one or even a few sentences. But once hundreds or thousands of sentences have to be anonymized this manual approach is no longer feasible. And it gets even more labor intensive if one would like to apply one of the "fancier" techniques like K-Anonymity etc. To deal with large data sets and complex anonymization tasks the process of identifying sensitive data and its subsequent anonymization needs to be automated.

To do this we can rely on a technique called Natural Language Processing (NLP). In short, NLP focuses on how computers understand, manipulate and generate natural language, such as written text [15]. Following this definition it can further be divided into Natural Language Understanding (NLU) and Natural Language Generation (NLG) [16]. As the name suggests NLU is mostly concerned about understanding language and extracting entities and meanings, while NLG is focused on generating replies [16]. Today NLP is used in many different applications, such as email spam detection, dialogue systems, summarization tasks, information extraction and many more [16].

However, as mentioned above our goal is not to detect spam or develop a chatbot, we "simply" would like to know which data is sensitive to anonymize it in a second step. Addressing the task of identifying and extracting specific information, such as keywords or entities from text is called Named Entity Recognition (NER), a specific technique of NLP and its "subclass" NLU [17].

4.4.1 Overview of Named Entity Recognition Approaches

Generally speaking NER can be achieved by applying four different approaches. Dictionary based NER, Rule based NER, Machine Learning based NER [18] and Deep Learning based NER [23]. In the following, we will quickly introduce the four different approaches. After that, we will present two chapters that focus specifically about NLP and NER in the context of Machine Learning and Deep Learning respectively.

4.4.1.1 Dictionary Based Named Entity Recognition

A dictionary based approach matches the word in a text against a dictionary and checks whether the word is part of the dictionary or not. While simple, it has severe limitations.

For instance, the dictionary needs to be constantly updated [25], as only words can be detected as entities that are part of the dictionary [26]. Another drawback is that a dictionary based approach might struggle when faced with words that vary in spelling [27]. With such drawbacks, the dictionary based NER does not seem to suffice for most tasks.

4.4.1.2 Rule Based Named Entity Recognition

While dictionary based NER matches words against a dictionary, rule based NER uses rules or patterns to extract entities [25]. For instance, one could extract names from a sentence by looking at all the words that are preceded with the prefix Dr. [27]. The drawback of such an approach is easy to recognize, since in the example above we would only identify names of people with a doctor title. Thus, creating extensive rules is time consuming and domain-dependent [26]. As such rule based methods might be applied to some specific use cases, but generally might be very limited in scope.

4.4.1.3 Machine Learning Based Named Entity Recognition

With supervised learning, NER can be interpreted as a multi-class labeling task [23]. That means we use annotated training data, consisting of an input and expected output to train a Machine Learning model [24]. Common techniques include Support Vector Machines, Hidden Markov Models, and Conditional Random Fields [24]. We will have a closer look at the Machine Learning based approaches in Section 4.5.

4.4.1.4 Deep Learning Based Named Entity Recognition

Compared to the more traditional Machine Learning based NER, Deep Learning based NER has three core strengths [23]. Deep Learning allows us to learn complex features from data via non-linear activation functions compared to the linear CRF or HMM [23]. Additionally, for traditional Machine Learning based NER, a lot of domain expertise is required, while Deep Learning based models are able to automatically learn useful representations from raw data [23]. Furthermore, with Deep Learning it is possible to train NER models end-to-end by applying gradient descent, making it possible to design complex NER systems [23]. We will have a closer look at how Deep Learning and Transformers can be leveraged to do NER and NLP in general in Section 4.6.

4.4.2 Using Identified Named Entities For Anonymization

Once we have gathered all the relevant entities by performing NER, either via Machine Learning or Deep Learning based NER, we can move on to the anonymization step. As discussed previously, performing anonymization manually for large amounts of data is not feasible. Thankfully, many tools exist that provide anonymization techniques to varying degrees as shown in Figure 4.13.

Privacy model	Model type	μ -Argus	sdcMi-cro	Open anonymizer	CAT	TIA-MAT	UTD	Anon	ARX	SEC-RETA	μ -Ant	Amnesia	Priopri-vacy	Probabilistic anonymization
k^m -Anonymity	Syntactic/statistical									X		X		
k -Anonymity	Syntactic/statistical	X ^a	X	X		X	X	X	X	X	X	X	X	
ℓ -Diversity	Syntactic/statistical				X		X	X	X					
t-Closeness	Syntactic/statistical				X		X		X		X			

Figure 4.13: Anonymization Tools [28]

A notable example on that list is ARX [30], as it is scalable, provides multiple anonymization techniques and allows risk assessments [28]. ARX accepts data in tabular form, which makes it easy to import our identified entities and perform the anonymization accordingly.

4.5 Machine Learning based NLP for Textual Data Anonymization

As we have previously discussed, in order to anonymize data, we first need to know what kind of data is sensitive. We can do this by applying various Machine Learning techniques, two of them will be discussed in this chapter. Such feature engineered supervised Machine Learning models are trained on inputs and their expected output to make predictions and can be used for NER [24]. Common examples are Support Vector Machines (SVM) and Conditional Random Fields (CRF) [24]. Once the entities have been identified we can make use of the various tools to anonymize data as shown in the previous section.

4.5.1 SVM

Support Vector Machines are used to solve binary classification problems. Assume we have a set of data points belonging to either class A or class B. The goal now is to find the optimal hyperplane that separates the two classes by maximizing the distance, also called margin, between the closest points of each class [31]. However, a big problem with this approach is its binary nature. That means that an SVM can only solve two-class problems, making it rather useless for classifying entities. One way to use SVMs to address n-class problems is by having multiple binary SVMs, each trained to distinguish class members from non-class members [29], or by using pair-wise-classification, with the goal to identify whether the sample should be classified as class A or B [29]. Obviously having multiple binary SVMs can create performance problems [29]. This is probably why it seems like they are seldom used these days.

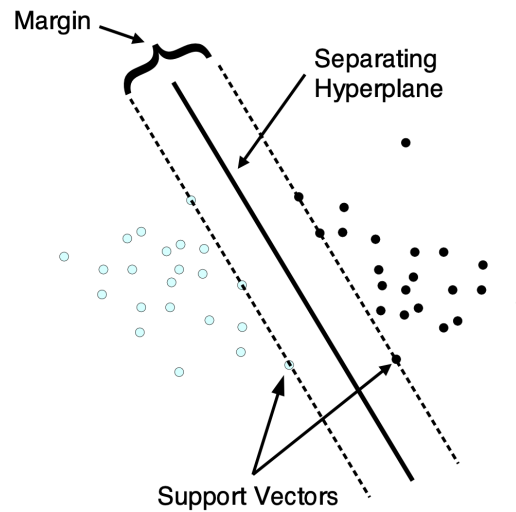


Figure 4.14: Anonymization Tools [31]

4.5.2 CRF

Conditional Random Fields are probabilistic graphical models that are often applied in Natural Language Processing tasks to extract Named Entities from a text [33]. CRFs are especially valuable for NER as they are able to take context into account. For instance, a linear CRF, predicts the Tag of a word based on its predecessor [34]. However, by only taking the previous word into account, Linear CRFs are severely limited in their usefulness. Possible ways to overcome this drawback are by introducing Bi-Directional-LSTMs and feeding its output into the CRF [34] or by making use of more General CRFs [32].

4.6 Deep Learning based NLP for textual Data Anonymization

When it comes to protecting sensitive data from disclosure, a Deep Learning based approach offers a different solution. As explained in section 4.4.1.4, this method does not require identifying the sensitive values in a dataset and then anonymizing them. Instead, Deep Learning models can learn which values should be anonymized and then perform the anonymization automatically. Deep Learning allows computers to recognize complex patterns and make predictions from large amounts of data. One popular type of a Deep Learning model is the Transformer, which was introduced in 2017 by Vaswani et al. [10].

In the realm of NLP, Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTMs) models were commonly used for sequence-to-sequence tasks [11]. This type of task involves transforming a given sequence of words in a sentence into another sequence [11]. While RNNs, with their recurrent connections, process the output sequence and only remember information from the previous n words, LSTMs perform better by incorporating

memory cells and gates to selectively store or discard information so that they decide which words from the input sequence to keep in memory at each step [11]. This makes LSTMs particularly adept at handling long-range dependencies between words in a sequence that are far apart, unlike RNNs which lack this capability [11]. However, as powerful as LSTMs are, they also face certain challenges. One major challenge is that LSTM's word-by-word processing can slow down the transformation process and that LSTMs struggle to capture dependencies that are too distant, leading to issues with long-term context understanding [11]. These limitations have inspired researchers to explore alternative architectures, leading to the rise of Transformer models.

Before delving into Transformers, it is worth mentioning the contribution of Large Language Models (LLMs). LLMs, such as the Large Language Model Meta AI (LLAMA) [42] and the ALPACA model [43], are advanced deep learning models with billions of parameters designed to comprehend and generate natural language and have demonstrated remarkable success in natural language tasks such as automatic summarization, machine translation, and question answering [41]. LLAMA was released by Meta AI in February 2023 and ranges from 7B to 65B parameters and ALPACA was released in March 2023 and has 7B parameters [42, 43]. Their achievements can be attributed to two key factors: the Transformer architecture with self-attention mechanisms that enhance their understanding of input relationships, and the utilization of self-supervised learning during pretraining, eliminating the need for manual annotation and allowing for efficient fine-tuning with minimal human-provided labels [41].

Transformers, which are a type of LLM, have emerged as a solution to the limitations of RNNs and LSTMs by using the attention mechanism to read all words in the input sequence in parallel, resulting in significantly faster computation [12]. Transformers are built on convolutional neural networks and consist of an encoder and a decoder stack. These two stacks are connected through attention [10]. These models are extensively used in NLP tasks such as translation, text summarization, question-answering, NER, and speech recognition [12]. The subsequent section will delve into the details of attention and the functioning of the encoder and decoder stacks.

4.6.1 Transformer Architecture

Transformers are a type of neural network architecture that has revolutionized the field of NLP [11]. They consist of an encoder and a decoder stack that work together to process input sequences, such as sentences, and produce corresponding output sequences [11, 12]. The key feature of Transformers is the attention mechanism, which allows the model to selectively focus on different parts of the input sequence depending on the context [11, 10, 12].

The input sequence is first transformed into a vector of embeddings in a high-dimensional space, where semantically similar words are positioned closer to each other [11, 12]. This vector is then fed into the encoder, which applies the attention mechanism to compute the relationship between different words in the sequence. The attention mechanism assigns weights to different parts of the input sequence based on their relevance to the current output, allowing the model to selectively focus on certain words while ignoring others

[11, 12]. By computing the relationships between different words in the sequence, attention helps the model to understand the context of each word and associate it with its proper meaning [11, 12]. These steps are repeated multiple times in parallel, with the attention mechanism attending to different parts of the sequence at each iteration [11, 10, 12].

To illustrate how the attention mechanism works, let's consider two similar sentences: "The cat drank the milk because it was hungry" and "The cat drank the milk because it was sweet" [12]. As shown in Figure 4.15 in the first sentence, the pronoun "it" refers to "cat," while in the second sentence, it refers to "milk." With the help of the attention mechanism, the Transformer model can more accurately associate the word "it" with the appropriate noun by considering the semantic meaning of each word in the input sequence [12].



Figure 4.15: Attention Mechanism [12]

The output of the encoder is then fed into the decoder, which uses a similar attention mechanism to generate the output sequence. The decoder outputs a probability distribution over the words in the vocabulary, and the word with the highest probability is selected as the predicted output for the next word in the sequence [11, 10, 12]. The output sequence of the Transformer model depends on the specific task assigned to the encoder. For example, if the encoder is given the task of translating a sentence, then the output would be the translated sentence. Alternatively, if the encoder is given the task of generating text, then the output would be the generated text. Another task that could be assigned to the encoder is anonymization, in which case the output would be an anonymized sequence.

Overall, the Transformer model's attention mechanism is a key feature that enables it to selectively attend to different parts of the input sequence based on their semantic meaning [11]. By capturing long-range dependencies between words in the input sequence without being limited by the sequential nature of RNNs and LSTMs, Transformers are highly effective for a wide range of NLP tasks, including machine translation, text summarization, question answering, NER, and speech recognition [11]. Therefore, Transformers have outperformed previous state-of-the-art models in many NLP tasks [11].

4.6.2 Training Transformers

After learning about the attention mechanism and the wide range of NLP tasks Transformers can handle, one may wonder how such powerful models are actually trained. The answer lies in a process called supervised learning, where the model is trained to predict the output sequence given an input sequence [11]. For instance, in machine translation tasks, pairs of source and target language sentences are used to train the model, with the input sequence being the source sentence and the output sequence being its corresponding translation [11]. By repeating this training process with many pairs of sentences, the model's parameters are optimized and its performance on unseen data is improved [11].

4.6.3 Generative Pretrained Transformer (GPT)

One particular implementation of Transformers that has gained significant attention in recent years is the Generative Pre-trained Transformer (GPT). These models are pre-trained on diverse datasets, enabling them to generate human-like responses to a variety of inputs [13]. They can then be fine-tuned for specific NLP tasks, such as language translation, question answering, and text summarization [13].

Model	Launch Date	Training Data	No. of Parameters	Max. Sequence Length
GPT-1	June 2018	Common Crawl, BookCorpus	117 million	1024
GPT-2	February 2019	Common Crawl, BookCorpus, WebText	1.5 billion	2048
GPT-3	June 2020	Common Crawl, BookCorpus, Wikipedia, Books, Articles, and more	175 billion	4096
GPT-4	March 2023	Unknown	Estimated to be in trillions	Unknown

Figure 4.16: GPT-1 to GPT-4 [13]

The first GTP, the GPT-1, was released by OpenAI in 2018, and had 117 million parameters. However, GPT-1 had a limitation in generating repetitive text when providing information beyond the scope of the training data [13]. The subsequent model, GPT-2, released in 2019, addressed this issue with an increased number of parameters, 1.5 billion, and a larger and more diverse training dataset [13]. GPT-2 was the first model to generate human-like responses, but it struggled with maintaining coherence in longer passages. These limitations were overcome by GPT-3, released in 2020, which boasted an enormous 175 billion parameters and a training dataset of about a trillion words as shown in Figure 4.16. GPT-3 was able to generate responses on NLP tasks even without explicit examples and could even write implementation code. One notable application of GPT-3 is the ChatGPT chatbot, which can generate human-like responses to any input. However, GPT-3's training dataset contained biased information, leading to biased responses, and it had limitations in fully understanding the context of a given prompt, sometimes leading

to irrelevant responses [13]. The most recent release, GPT-4, launched on March 14, 2023, overcomes these limitations and offers even more improvements. For instance, GPT-4 can interpret images as text prompts, expanding the range of inputs it can process [13].

4.6.4 ChatGPT for Data Anonymization

With the recent advancements of GPTs and ChatGPT especially, the question arises whether or not these tools are useful for anonymization tasks. We will discuss the possibilities and problems of using GPTs on the basis of ChatGPT (Version: Mar 2023).

In a perfect world, an anonymization tool would receive the raw data, and maybe some prompts about what should be anonymized. Based on that, it would produce an accurate and complete result. This promises a very easy approach to data anonymization. And at first glance, it seems like ChatGPT might be able to achieve this. Getting started with ChatGPT is easy, as the only prerequisite needed is a computer with an internet connection and an account on the website of OpenAI (<https://openai.com>). After that, one is already set up and ready to start anonymizing. As an example, we asked ChatGPT the following very easy prompt: *"Anonymize the following sentence: Peter Parker is suffering from a spider bite."* The prompt resulted in the output shown in Figure 4.17.

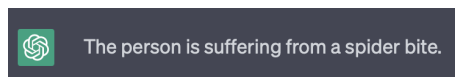


Figure 4.17: Anonymization of a sentence

This example nicely shows, how easy it is to use ChatGPT for such tasks. Furthermore, it also highlights that users do not have to think about NER at all. However, we have to consider more difficult examples. As such we decided to replicate the K-Anonymity example from earlier. We asked ChatGPT to create a table with the same data as in Figure 4.11a. After it created the original table correctly we then asked ChatGPT the following: *"Apply K-Anonymity with $K=4$ on the table."* This resulted in the anonymized table which is shown in Figure 4.18.

However, the result of this table is not K-Anonymous with $K=4$. This shows that, in its current form, ChatGPT can still make mistakes when faced with data anonymization tasks. Furthermore, when comparing the resulting table shown in Figure 4.18 to the previously discussed table shown in Figure 4.11b, it is immediately obvious that the tables differ. That is due to the choice of grouping and generalization used. Interestingly, even ChatGPT itself notes that there are different ways to anonymize the table as can be seen in Figure 4.19.

While it might be possible (but not guaranteed due to ChatGPT's non-determinism) to arrive at the exact same (and correct) output with proper prompt engineering, we have not been able to replicate it for this example. The problem of prompt engineering and ChatGPT's non-deterministic nature, which results in varying replies even if the exact same question is asked [35], is something to be aware of. This definitely can be a drawback, especially when compared to the more traditional Machine Learning approach

Zip Code	Age	Nationality	Condition
1305*	20-29	Russian, American	Heart Disease
1306*	20-29	American	Heart Disease
1306*	20-29	American	Viral Infection
1305*	20-29	American	Viral Infection
1305*	50-59	Indian, Russian	Cancer
1305*	50-59	Indian, Russian	Heart Disease
1305*	40-49	American	Viral Infection
1305*	40-49	American	Viral Infection
1305*	30-39	American	Cancer
1305*	30-39	Indian	Cancer
1306*	30-39	Japanese, American	Cancer
1306*	30-39	American	Cancer

Figure 4.18: ChatGPT K-Anonymization

In this anonymized table, we have grouped together rows that have the same first digit in the Zip Code column (i.e., 1305 or 1306), and generalized the values of the Age and Nationality columns to ranges and combinations that occur in at least 4 rows. The Condition column remains unchanged, as it is not a quasi-identifying attribute.

Note that there are different ways to apply K-Anonymity, and the choice of grouping and generalization depends on the specific context and requirements of the anonymization task.

Figure 4.19: ChatGPT Reasoning

with subsequent use of a dedicated anonymization tool, which can be made deterministic. Theoretically, one could also try to make a Transformer, such as ChatGPT, deterministic. For that Xu, Xiangzhe, et al. differentiate between two sources of non-determinism: algorithmic and non-algorithmic [44]. The algorithmic source describes the randomness that developers intentionally introduce, such as dropout layers and adding noise. This could potentially be addressed by setting fixed random seeds [44]. Contrary to the algorithmic source, the non-algorithmic source describes randomness that developers can not control. An example of a non-algorithmic source is data parallelism in Multi-Threading. An in-depth discussion of these sources and how one might potentially achieve determinism can be found in the paper "Checkpointing and Deterministic Training for Deep Learning" by Xu, Xiangzhe, et al [44]. In addition to non-determinism we also need to consider that answers might be factually wrong or contain prejudices [37]. It seems obvious that such errors can also affect anonymization tasks, as we have seen in our example when trying to achieve K-Anonymity. Another important aspect to consider is the legality of using ChatGPT for anonymization. As anonymization is usually concerned with sensitive data, it might not be allowed to call external services such as ChatGPT. That can be due to Data Privacy Regulations or even domain specific regulations [38] such as the Health Insurance Portability and Accountability Act [36].

4.7 Discussion and Conclusion

With this paper we provided an overview of how textual data can be anonymized either by relying on Machine Learning models and anonymization tools or by using more "modern" approaches like ChatGPT.

While the later seems like a dream come true for data anonymization, it still faces various challenges rooted in its non-determinism, errors, trained prejudices, and data privacy concerns. However, we think that if these problems can be addressed the future of GPTs for textual data anonymization can be bright. This is especially true when considering that they can work as a single anonymization tool. Comparing this to the more traditional approach with Machine Learning based NER and subsequent use of anonymization tools, it makes for a quicker, easier, and more streamlined solution. However, the traditional approach has various benefits as well, as it can easily be made deterministic and is able to address the privacy concerns that come with the use of ChatGPT.

As part of this paper, we did not perform an in-depth statistical comparison to decide which approach is better. However, our paper motivates further research in that area. It might be interesting to quantitatively compare error rates of the traditional Machine Learning approach and ChatGPT. Additionally, one area of interest could be ChatGPTs anonymization capabilities, i.e. which anonymization models are supported etc. In that area future research could use our consolidated overview of anonymization techniques and models in Section 4.2 as a starting point.

Bibliography

- [1] Garfinkel, S. et al.: De-identification of Personal Information. National Institute of Standards and Technology (NIST), US Department of Commerce, National Institute of Standards and Technology, 2015.
- [2] Label Your Data: Data Anonymization Tools for Machine Learning and AI. <https://labelyourdata.com/articles/data-anonymization-tools-for-machine-learning-and-ai>, accessed April 16, 2023.
- [3] Sweeney, L.: k-anonymity: A model for protecting privacy; *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, Vol. 10, No. 5, pp. 557–570, 2002, Publisher: World Scientific.
- [4] Hamza, N. and Hefny, H.A. et al.: Attacks on anonymization-based privacy-preserving: a survey for data mining and data publishing; *Publisher: Scientific Research Publishing*, 2013.
- [5] Machanavajjhala, A., Kifer, D., Gehrke, J., and Venkitasubramaniam, M.: l-diversity: Privacy beyond k-anonymity; *ACM Transactions on Knowledge Discovery from Data (TKDD)*, Vol. 1, No. 1, pp. 3-es, 2007, Publisher: ACM New York, NY, USA.
- [6] Dwork, C.: Differential privacy; In *Automata, Languages and Programming: 33rd International Colloquium, ICALP 2006, Venice, Italy, July 10-14, 2006, Proceedings, Part II 33*, pp. 1–12, Springer, 2006.
- [7] Li, N., Qardaji, W., and Su, D.: On sampling, anonymization, and differential privacy or, k-anonymization meets differential privacy; In *Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security*, pp. 32–33, 2012.
- [8] R. Bild, K. A. Kuhn, and F. Prasser, "SafePub: A Truthful Data Anonymization Algorithm With Strong Privacy Guarantees," *Proceedings on Privacy Enhancing Technologies*, vol. 2018, no. 1, pp. 67–87, 2018.
- [9] ARX De-identifier: Privacy Criteria - ARX; <https://arx.deidentifier.org/overview/privacy-criteria/>, [Online; accessed 20-April-2023].

- [10] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.
- [11] Joshi, S. and Modi, S.: What is a Transformer? <https://medium.com/inside-machine-learning/what-is-a-transformer-d07dd1fbec04>, April 2019, [Online; accessed April 24, 2023].
- [12] Jay Alammar: Transformers Explained Visually (Part 1): Overview of Functionality. <https://towardsdatascience.com/transformers-explained-visually-part-1-overview-of-functionality-95a6dd460452>, August 2018, [Online; accessed 24-April-2023].
- [13] Proctor, B.: GPT Models Explained and Compared: Which One Is Best for You?; [https://www.makeuseof.com/gpt-models-explained-and-compared/#:~:text=Generative%20Pre%2Dtrained%20Transformers%20\(GPTs,relevant%20and%20semantically%20coherent%20language](https://www.makeuseof.com/gpt-models-explained-and-compared/#:~:text=Generative%20Pre%2Dtrained%20Transformers%20(GPTs,relevant%20and%20semantically%20coherent%20language), July 2021, [Online; accessed 24-April-2023].
- [14] TechTarget: GPT-3 (Generative Pre-trained Transformer 3); <https://www.techtarget.com/searchenterpriseai/definition/GPT-3>, 2021, [Online; accessed on 25-April-2023].
- [15] Chowdhary, KR1442 and Chowdhary, KR: Natural language processing; Fundamentals of Artificial Intelligence, Springer, 2020, pp. 603–649.
- [16] Khurana, Diksha, Koli, Aditya, Khatter, Kiran, and Singh, Sukhdev: Natural language processing: State of the art, current trends and challenges. *Multimedia tools and applications*, vol. 82, no. 3, Springer, 2023, pp. 3713–3744.
- [17] Marrero, Monica, Urbano, Julian, Sanchez-Cuadrado, Sonia, Morato, Jorge, and Gomez-Berbs, Juan Miguel: Named entity recognition: fallacies, challenges and opportunities; Computer Standards and Interfaces, vol. 35, no. 5, Elsevier, May, 2013, pp. 482–489.
- [18] Hye-Jeong Song, Byeong-Cheol Jo, Chan-Young Park, Jong-Dae Kim, and Yu-Seop Kim. Comparison of named entity recognition methodologies in biomedical documents; Biomedical Engineering Online, vol. 17, no. 2, BioMed Central, 2018, pp. 1–14.
- [19] GDPR: Info (Complete guide to GDPR compliance); <https://gdpr.eu>, 2023, [Online; accessed on 25-April-2023].
- [20] GDPR (Territorial scope); <https://gdpr-info.eu/art-3-gdpr/>, 2023, [Online; accessed on 25-April-2023].
- [21] AI Multiple: Data Anonymization - Definition, Techniques and Tools; Website, <https://research.aimultiple.com/data-anonymization/>, 2023, [Online; accessed on 28-April-2023].

- [22] B. Medlock: An Introduction to NLP-based Textual Anonymisation; Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC), Genoa, Italy, May 2006, pp. 1051-1056.
- [23] J. Li, A. Sun, J. Han, and C. Li: A survey on deep learning for named entity recognition; *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 1, pp. 50-70, 2020.
- [24] JYadav, Vikas and Bethard, Steven: A Survey on Recent Advances in Named Entity Recognition from Deep Learning models; arXiv preprint arXiv:1910.11470, 2019.
- [25] Turing: A Comprehensive Guide to Named Entity Recognition; <https://www.turing.com/kb/a-comprehensive-guide-to-named-entity-recognition>, [Online; accessed on 28-April-2023].
- [26] T. Eftimov, B. Korousic Seljak, and P. Korosec: A rule-based named-entity recognition method for knowledge extraction of evidence-based dietary recommendations; *PLOS ONE*, vol. 12, no. 6, pp. 1-32, June 2017, <https://doi.org/10.1371/journal.pone.0179488>.
- [27] TechTarget: Definition - named entity recognition (NER); <https://www.techtarget.com/whatis/definition/named-entity-recognition-NER>, April 28, 2023.
- [28] Haber, A. C., Sax, U., Prasser, F., and the NFDI4Health Consortium. (2022). Open tools for quantitative anonymization of tabular phenotype data: literature review. *Briefings in Bioinformatics*, 23(6), bbac440. <https://doi.org/10.1093/bib/bbac440>
- [29] Hideki Isozaki, Hideto Kazawa. Efficient Support Vector Classifiers for Named Entity Recognition *COLING 2002: The 19th International Conference on Computational Linguistics*, 2002
- [30] Arx - A Comprehensive Tool for Anonymizing Biomedical Data; <https://arx.deidentifier.org>, [Online; accessed on 28-April-2023].
- [31] Meyer, D. and Wien, F.T.: Support vector machines. *R News*, vol. 1, no. 3, pp. 23-26, 2001.
- [32] Sutton, Charles and McCallum, Andrew and others: An introduction to conditional random fields. *Foundations and Trends in Machine Learning*, vol. 4, no. 4, pp. 367-373, 2012.
- [33] John Snow Labs: Step-by-Step Tutorial Conditional Random Fields for Named Entity Recognition in Python; <https://www.johnsnowlabs.com/step-by-step-tutorial-conditional-random-fields-for-named-entity-recognition-in> [Online; accessed on 28-April-2023].
- [34] Domino Data Lab: Named Entity Recognition (NER): Challenges and Models; <https://www.dominodatalab.com/blog/named-entity-recognition-ner-challenges-and-model>, [Online; accessed on 28-April-2023].

- [35] Toolonomy: ChatGPT FAQs: Unusual Questions and Answers About the AI Chatbot; <https://toolonomy.com/chatgpt-faqs/>, [Online; accessed on 28-April-2023].
- [36] CDC: HIPPA (Health Insurance Portability and Accountability Act of 1996 (HIPAA)); <https://www.cdc.gov/phlp/publications/topic/hipaa.html>, 1996, [Online; accessed on 25-April-2023].
- [37] Zhang, M. and Li, J.: A commentary of GPT-3 in MIT Technology Review 2021, Fundamental Research, vol. 1, no. 6, pp. 831-833, 2021, Elsevier.
- [38] Chintagunta, Bharath and Katariya, Namit and Amatriain, Xavier and Kannan, Anitha: Medically aware GPT-3 as a data generator for medical dialogue summarization. *Machine Learning for Healthcare Conference*, pp. 354-372, 2021.
- [39] GDPR: Recital 26, Not Applicable to Anonymous Data; <https://gdpr-info.eu/recitals/no-26>, [Online; accessed on 24-May-2023].
- [40] SAGE Journals: "Exploring the Impact of Artificial Intelligence on Society and Education: A Review"; *Research on Education and Media*, (Volume 51, Issue 2), May 2021, pp. 223-240, <https://journals.sagepub.com/doi/full/10.1177/00333549211026817>, [Online; accessed on 31-May-2023].
- [41] Y. Shen, L. Heacock, J. Elias, K. D. Hentel, B. Reig, G. Shih, L. Moy: "ChatGPT and other large language models are double-edged swords"; *Radiology*, (Volume 307, Issue 2), e230163, 2023, Radiological Society of North America.
- [42] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, and others: "Llama: Open and efficient foundation language models"; *arXiv preprint arXiv:2302.13971*, 2023.
- [43] Stanford Center for Reservoir Forecasting and Management: "Alpaca: Advances in Large-scale Probabilistic AI for Complex Applications"; *CRFM Blog*, March 13, 2023, <https://crfm.stanford.edu/2023/03/13/alpaca.html> [Online; accessed on 31-May-2023].
- [44] Xu, Xiangzhe and Liu, Hongyu and Tao, Guanhong and Xuan, Zhou and Zhang, Xiangyu: "Checkpointing and deterministic training for deep learning"; *Proceedings of the 1st International Conference on AI Engineering: Software Engineering for AI*, pp. 354-372, 2022. <https://www.cs.purdue.edu/homes/taog/docs/CAIN22.pdf> [Online; accessed on 30-May-2023].

Chapter 5

Security Concerns of Privacy-preserving and Decentralized Artificial Intelligence

Huiran Duan, Han Yang

In recent years, ensuring security and privacy in federated learning (FL) has become increasingly crucial. However, research in this field is still limited, and the issue of adversarial attacks in distributed federated learning (DFL) is largely unexplored. In response to this, we present a literature review on CFL and DFL, covering the most recent attacks and defenses. Our work answers several research questions, including the fundamentals and differences between CFL and DFL, recent attacks in CFL, and the connections and differences between CFL and DFL in the face of attacks. We also examine popular defense methods in CFL and discuss their applicability and adaptation to DFL. Our findings provide valuable insights for researchers and practitioners seeking to secure FL systems.

Contents

5.1	Introduction	99
5.2	Motivation and Contributions	100
5.3	Fundamentals of Federated Learning	102
5.3.1	Centralized Federated Learning	103
5.3.2	Decentralized Federated Learning	106
5.4	Attacks in CFL	108
5.4.1	Client-side attacks in CFL	108
5.4.2	Server-side attacks in CFL	111
5.4.3	Federation-side attacks in CFL	111
5.5	Attacks in DFL	114
5.5.1	Connections between attacks in CFL and DFL	115
5.5.2	Vulnerabilities of DFL Facing Attacks	116
5.6	Defense Methods in CFL	119
5.6.1	Cryptographic approaches	121
5.6.2	Aggregation algorithms	124
5.6.3	Client Identification	125
5.6.4	Model Training	126
5.7	Defense Methods in DFL	127
5.7.1	Defense methods inherited from CFL ecosystem	127
5.7.2	Defense methods not applicable to DFL ecosystems	128
5.7.3	Unique defense methods in DFL ecosystem	129
5.8	Open Challenges & Future Directions	130
5.9	Conclusion	131

5.1 Introduction

Artificial Intelligence (AI) is a dynamic and ever-evolving field that aims to create machines capable of performing tasks that traditionally require human intelligence. AI and, in particular, Machine learning (ML) has been propelled forward by the abundance of data available in recent years [1]. With phones and tablets becoming primary computing devices [2] for many people, the powerful sensors on these devices - including cameras, microphones, and GPS - provide an unparalleled amount of private data [3]. Most current AI systems are centralized, relying on a single central entity to store and process data. In this data-centered approach, participants, which can be smartphones, watches, hospitals, and so on, send their data to a central server where the data is processed through typical workflows, including ML modeling, training, evaluation, and deployment. However, in practice, limited communication resources, data privacy, or specific regulations often make data collection in central entities inefficient or impractical [29], as done in traditional ML. These limitations have led to the emergence of a new approach named Federated Learning (FL) [42] to train ML models, which has gained significant momentum in recent years. FL enables the development of AI models without the need to centralize data, making it possible to train models locally across individual devices. This approach helps preserve data privacy and avoids the limitations of a centralized architecture, which can be applied to various scenarios, such as healthcare, industry 4.0, mobile devices, military, and vehicles [29].

FL can be classified into two categories according to how federated models are created: centralized, known as Centralized Federated Learning (CFL), and decentralized, known as Decentralized Federated Learning (DFL) [54]. Presently, the CFL approach is the most widely used method in FL [29]. It employs a central orchestration server to generate and distribute a global model to all the participating clients. In detail, clients train their models with local data and transmit the local model updates to a central server [42]. The server then learns a global model by aggregating and combining the individual model updates. Despite the advantages of CFL, it faces several core challenges [44, 4], including 1) high communication costs, 2) system and statistical heterogeneity, 3) a bottleneck in the central server, and 4) the existence of a single point of attack.

DFL is designed to mitigate certain challenges posed by CFL, particularly the bottleneck effect of a central server and the susceptibility of a single point of attack [29]. In some other literature, Decentralized Federated Learning (CFL) is referred to as Distributed, Serverless, or Peer-to-Peer Federated Learning. However, for the purposes of this report, we will refer to it as DFL. It distinguishes itself from CFL by operating without a centralized server to manage the learning process [54]. Consequently, there is no central server within the network, and each node participates by exchanging model updates with neighboring nodes, allowing each node to act as an aggregator. Although DFL addresses the issues of a central server, it introduces new challenges [29] such as the absence of a powerful central server to coordinate the training process and the need to confront additional attacks in distributed networks.

Current literature on attacks and defense methods in Federated Learning (FL) predominantly focuses on the Centralized Federated Learning (CFL) domain, with limited research on the corresponding attacks and necessary defense strategies for Decentralized

Federated Learning (DFL). Therefore, this report aims to compare the fundamentals of CFL and DFL and analyze their distinct vulnerabilities when subjected to various types of attacks, including a detailed analysis of the typical attacks faced by each. Additionally, this report investigates the state-of-the-art defense mechanisms that can be applied to each approach, while scrutinizing the security and privacy issues associated with these two types of federated learning.

Finally, we conclude our comparison and analysis of CFL and DFL and outline some of the open challenges and future research directions. One of the critical issues that require further investigation is how to strike a balance between privacy, security, and performance. Also, while stronger privacy protections are desirable, they may come at the cost of reduced model accuracy. Therefore, it is crucial to develop techniques that can guarantee privacy while preserving the model’s utility. Additionally, the use of privacy-preserving techniques can impose additional computational and communication costs on the federation, making scalability a significant challenge. Hence, we can focus future work on reducing these overhead costs while maintaining an adequate defense against attacks.

5.2 Motivation and Contributions

In recent years, federated learning (FL) has become an increasingly popular technique in machine learning research [44]. However, ensuring security and privacy in FL remains a challenging problem [61]. There is a strong need to ensure that the data and models used are secure and private because of the growing use of FL in many application situations. When sensitive data is involved, as in the medical and financial industries, the requirement for security and privacy is especially crucial. Despite the fact that security and privacy are becoming increasingly important in FL research, there hasn’t been much systematic investigation into these topics. In particular, the issue of adversarial attacks in Distributed Federated Learning (DFL) is still largely unexplored. Adversarial attacks can severely compromise the security and privacy of the data and models being used in DFL. This demonstrates the necessity of conducting additional studies in this field in order to create strong defenses against such assaults.

While numerous studies have investigated security and privacy issues in centralized machine learning [12], systematic research on FL privacy and security is still relatively scarce. Moreover, the emergence of decentralized federated learning (DFL) has added new dimensions to the problem [29], as it requires further consideration of the topologies and peer-to-peer communications in a distributed system. Although there have been some attempts to address privacy-preserving techniques for CFL [5, 7], the literature is still limited in terms of how these methods can be adapted to the DFL. To address this gap, we present a comprehensive literature review that focuses on both CFL and DFL, and provides an overview of the most recent and representative attacks and defenses in these two settings. In this context, this report identifies several open research questions that require investigation:

RQ1. What are the fundamentals of CFL and DFL, and what are the differences? In order to analyze how attackers exploit vulnerabilities in CFL and DFL, a comprehensive

Table 5.1: Research questions in this report

Research question	Aim	Section
RQ1. What are the fundamentals of CFL and DFL, and what are the differences?	To gain a comprehensive understanding of their fundamentals	5.3
RQ2. What are the most recent and representative attacks in CFL?	To conduct a thorough investigation of attacks in CFL	5.4
RQ3. How do those attacks affect DFL, and what are the connections and differences?	To compare the attacks encountered by DFL with those faced by CFL	5.5
RQ4. What are the popular defense methods in CFL?	To provide a comprehensive understanding of the popular defense methods in CFL	5.6
RQ5. Are these defense methods applicable to DFL, and how are they applied?	To gain a deeper understanding of the relationship between defense methods used in CFL and DFL	5.7

understanding of their fundamentals is necessary. This entails examining common approaches and challenges associated with both techniques. Although several variants of CFL and DFL have been proposed in the literature, a detailed and comprehensive comparison between the two is lacking. Additionally, to evaluate how CFL and DFL differ in terms of facing attacks and implementing defenses, it is imperative to compare and analyze their respective strengths and weaknesses individually.

RQ2. What are the most recent and representative attacks in CFL? It is necessary to begin by conducting a thorough investigation of attacks in CFL, which can serve as the basis for subsequent comparisons with those in DFL. Additionally, the existing taxonomy of attacks in the literature does not clearly highlight the distinctions between CFL and DFL attacks. Thus, it is necessary to propose a taxonomy that enables more straightforward comparisons between the two types of FL.

RQ3. How do those attacks affect DFL, and what are the connections and differences? It is necessary to compare the attacks encountered by DFL with those faced by CFL. Specifically, we must analyze which attacks DFL can prevent compared to CFL, and which attacks pose a greater threat to DFL. Additionally, it is essential to determine the vulnerabilities of DFL that attackers exploit to make it more susceptible to these attacks.

RQ4. What are the popular defense methods in CFL? To provide a comprehensive understanding of the popular defense methods in CFL, this section aims to classify them in a reasonable way and compare their effectiveness in mitigating attacks. Furthermore, we will also discuss the drawbacks of each method separately to provide a balanced perspective.

RQ5. Are these defense methods applicable to DFL, and how are they applied? To gain a deeper understanding of the relationship between defense methods used in CFL and DFL, we aim to explore the popular defense methods in CFL in RQ4. Additionally, we

aim to investigate whether these defense methods are applicable in DFL, and if so, how they can be modified to suit DFL.

These research questions in 5.1 are interrelated and follow a logical sequence to provide a comprehensive comparative analysis of CFL and DFL regarding security and privacy. The analysis of the fundamentals of both techniques serves as the foundation of this report, while the focus is on the approaches to attack and defense, and the comparative analysis in CFL and DFL. To answer the previous research questions, the main contributions of this report are:

- A comparative analysis of the fundamentals of CFL and DFL, encompassing common approaches, challenges, and advantages of each technique (answering RQ1 in Section 5.3).
- An overview of attacks in CFL, with a taxonomy based on attack location, including client-side, server-side, and federation-side attacks (answering RQ2 in Section 5.4).
- An analysis of the connections and differences between attacks in DFL and CFL, with explanations of why DFL is more susceptible to certain attacks, such as malware, poisoning, free-riding, Sybil, and eclipse attacks (answering RQ3 in Section 5.5).
- An overview of defense methods in CFL and classified based on technical characteristics of the defense methods and their purpose of the application, including Cryptographic approaches, Aggregation algorithms, Client Identification, and Model Training (answering RQ4 in Section 5.6).
- An analysis of connections and differences of defense methods in DFL and CFL, whether they are applicable, and how to modify them to suit DFL(answering RQ5 in Section 5.7).

5.3 Fundamentals of Federated Learning

Federated Learning (FL) allows for training models on decentralized data sources, without the need for data collection into some central entities. In FL, data is distributed across multiple devices, and each device trains a local model using its own data. FL can be classified into two main types: Centralized Federated Learning(CFL) and Decentralized Federated Learning(DFL). Both CFL and DFL face the challenges of the system and statistical heterogeneity [36, 37] and expensive communication [44]. Depending on the network topology of the DFL, it may even increase the network overhead.

In CFL, a central server coordinates the training process. The central server collects the local model updates from each device, aggregates them, and sends the updated global model back to the devices for further training [40]. The fundamental principle of CFL is that the central server acts as a coordinator for the training process, and all devices communicate directly with the server. CFL is useful when the devices have limited computational power and can only perform simple tasks. However, there are several challenges

in CFL, such as the need for a trusted and reliable server and it presents a communication bottleneck [41] as it exchanges information with every node.

In DFL, devices communicate with each other directly, without a central server. In DFL, devices collaborate to train models without a central coordinator [8]. The fundamental principle of DFL is that each device shares the model updates with other devices in the federation network. The devices then use the shared updates to update their local models. DFL has several advantages compared to CFL, such as higher fault tolerance, more computing power, better scalability, improved privacy, and no central server bottleneck nor a single point of attack.

5.3.1 Centralized Federated Learning

In most of the literature, CFL is often referred to directly as FL. One reason is that FL was first proposed as the centralized paradigm [42]; another reason is that CFL predominates both in research and application compared to DFL [29]. CFL assumes a central server that coordinates the training process. The most fundamental difference between this and data-centered machine learning is that it allows local data to be retained locally among the participants. However, they are also related in that both of them require a trusted central server with which each node in the network communicates. Therefore, although CFL is designed to address the issue of central data leakage, the problems presented by a central server remain [28]. In this chapter, we review the fundamentals and challenges of CFL.

5.3.1.1 Problem formulation

The primary objective of standard federated learning is to learn a global statistical model from data that is distributed across a large number of remote devices, ranging from tens to millions. The aim is to accomplish this while ensuring that the data generated by the devices remain stored and processed locally, with only intermittent updates being communicated to a central server. Thus, we can write the typical objective function of CFL as

$$\min_w F(w), \text{ where } F(w) = \sum_{k=1}^m p_k F_k(w) \quad (5.1)$$

In this context, m represents the overall count of devices, and p_k is a non-negative value such that the sum of all p_k values equals one. F_k denotes the local objective function associated with the k th device. Typically, the local objective function is defined as the empirical risk over local data. This is expressed as $F_k(w) = \frac{1}{n_k} \sum_{j_k=1}^{n_k} f_{j_k}(w; x_{j_k}, y_{j_k})$, where n_k represents the number of samples available locally. The user-defined term p_k indicates the relative impact of each device, with two common settings being $p_k = \frac{1}{m}$ or $p_k = \frac{n_k}{n}$, where n represents the total number of samples. While we will refer to (5.1) in this article, it's worth noting that other objectives or modeling approaches may be more suitable depending on the specific application [44].

5.3.1.2 Common Approach of CFL

The common approach of Centralized Federated Learning (CFL) involves a series of steps that enable a central server to coordinate the learning process [44]:

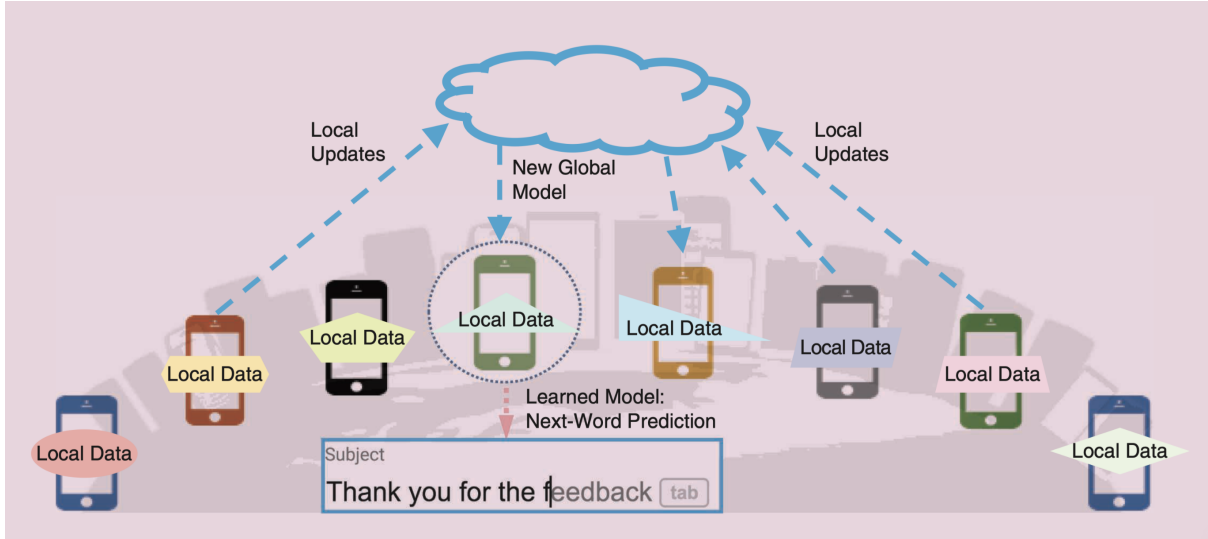


Figure 5.1: A demonstration of common CFL approaches

Step 1 in CFL is for the server to send an initial model to all participating clients. This model serves as a starting point for the clients to train their local models on their respective datasets. The initial model is typically a pre-trained model or a random initialization [42]. Before the training process begins, the server sets up the training hyperparameters, such as the number of training epochs, batch size, and learning rate, and shares them with all participating clients.

In step 2, the clients perform the training process on their local datasets using stochastic gradient descent (SGD) or its variations such as the Qsparse-local-SGD algorithm [43], which employs a blend of aggressive sparsification, quantization, and local computation, accompanied by error compensation. Then the clients send the computed model updates to the server for aggregation.

In step 3, the server receives the model updates from the clients and aggregates them using a specific aggregation method. The aggregation method can vary depending on the application and the problem being addressed. For instance, Federated Averaging (FedAvg) [42] is a widely used aggregation method that averages the model updates received from the participating clients. Other aggregation methods, such as FedProx [44], take into account the magnitude of the updates or the performance of the clients.

After the server aggregates the model updates, it updates the shared model in step 4. The updated global model is then sent back to the clients for the next round of training. This cycle of steps 2,3 and 4 is repeated iteratively until the global model reaches a performance criterion.

5.3.1.3 Challenges in CFL

In order to optimize the objective function (5.1), CFL faces some core challenges in its process. Some of these challenges are unique to FL, such as expensive communication and severe heterogeneity [41], and others are due to the central server, such as the central server bottleneck and a single point of attack [44].

Expensive communication In CFL, data transmissions can become expensive with a large number of participating nodes. This is particularly true for scenarios with millions of smartphones, where high network overhead and latency can arise from transmitting a large number of model parameters many times. To reduce the size of the transmitted messages at each round, researchers have proposed several approaches mainly based on compression techniques. And a variant approach of CFL is allowed to select a subset [47] of devices by the server for each communication round based on various criteria, such as reward [45], reputation [48, 56], probabilities [46], computational power, and connectivity of devices [52]. To reduce the total number of communication rounds, local participants can perform multiple model update computations [49] before transmitting them to the central server. However, the number of local computations is a hyper-parameter that introduces new challenges to the aggregation algorithm of the central server when the number is too large.

Systems and statistical heterogeneity Participants in CFL often have different hardware and network capabilities, which can introduce problems such as stragglers [53]. Devices may drop out at a given iteration due to factors such as poor CPU and memory, network connectivity, and battery level. In addition, data across devices in CFL may be unbalanced [51], with a significant variation in the number of data samples. Furthermore, the data in the federation are usually non-IID [50], meaning that any particular local dataset is not representative of the population distribution due to variations among user datasets. To address these issues, more efficient aggregation algorithms are required to ensure convergence, and the asynchronous CFL [75] can also alleviate the problem of stragglers and non-IID data.

Central server bottleneck In CFL, the central server plays a crucial role in exchanging information with every node. However, this centralized approach can become a bottleneck, especially when the number of participating nodes increases, which can affect the scalability and efficiency of the system.

A single point of attack Another challenge in CFL is the risk of leaking sensitive information when there is no trust in the central server node. This can be particularly problematic in scenarios such as healthcare, where the data being shared is highly sensitive.

5.3.2 Decentralized Federated Learning

In contrast to the CFL's star network topology, DFL does not have a central node in the network, which implies a fundamental difference between DFL and CFL so that DFL does not rely on a central server to coordinate the learning process and each node can play the role of an aggregator [duan] [duan]. Such a network topology inherits many of the benefits of a distributed network by design, but at the same time poses some challenges in the context of federated learning. We will briefly describe the common approach of DFL and compare its challenges and advantages with CFL.

5.3.2.1 Common Approach of DFL

Algorithm 1 is a common approach of DFL, which outlines the steps involved in training a machine learning model in a distributed network.

Algorithm 1 A simplified algorithm of DFL

```

step 1: initialize local models
for each communication round do
  for each selected node in parallel do
    step 2: Computes local updates with its local data
    step 3: Exchanges updates with neighboring nodes
    step 4: Aggregates all updates and updates its local model
  end for
  if model reaches a performance criterion then
    break
  end if
end for

```

The same as in CFL, step 1 is to initialize local models for each participant. When a new node joins the network, it needs to get a copy of the current model to start contributing to the training process. After that, for each communication round and each selected node in parallel, it computes the local model updates with its local data. Step 2 is the fundamental difference between DFL and CFL, where each node exchanges model updates with neighboring or available nodes rather than a central server. The selection of nodes can be based on various criteria, such as proximity [57], network connectivity, or node reputation [56]. In step 3, each node aggregates all the model updates it has received and updates its local model. The aggregation process can be performed using various techniques [54, 55].

5.3.2.2 Challenges of DFL Compared to CFL

As discussed before, the CFL faces some challenges, and the DFL has emerged to address some of these challenges. Table 5.2 shows the differences between CFL and DFL when facing these challenges. The distributed nature of DFL eliminates the need for a central

server, which helps to overcome the challenges associated with a single point of attack and server bottleneck. Instead of relying on a central server, DFL distributes the training process across multiple nodes in the network. However, DFL still faces challenges related to systematic and statistical heterogeneity, a challenge that is an inherent property in federations and for which distributed networks do not provide significant improvements. Furthermore, DFL can even increase the network overhead. In some specific network topologies, such as a fully connected network topology, the number of communication connections among nodes increases exponentially with the number of nodes.

Table 5.2: Comparison of challenges in CFL to those in DFL

Challenges	In CFL	In DFL
Central server bottleneck	Yes	No
Single point of attack	Yes	No
Systems and Statistical Heterogeneity	High	High
Communication cost	Expensive	More Expensive (according to the network topology)

5.3.2.3 Advantages of DFL over CFL

Besides avoiding the challenges associated with a central server, DFL has some other advantages. Its high fault tolerance and self-scaling features [58] are inherited from the distributed network. It also increases computing power in general, depending on the topology of the distributed network and its number of nodes.

Higher Fault Tolerance In a DFL system, participating devices continuously update their knowledge about available nodes in the network instead of relying on a single centralized server. This makes DFL systems resilient to potential issues such as participating devices disappearing, straggling, or reappearing in the network [54]. Even if some of the nodes become unavailable, the overall system can still function as long as there are other nodes available to contribute to the training process.

Self-scaling DFL systems are designed to be decentralized, and this makes them self-scaling [29]. When a new node joins the network, other nodes can begin communicating with it, considering factors such as device throughput or computational capacity, without requiring additional resources on a central server. This means that DFL systems can scale efficiently, which is essential for applications that require a large number of participating devices, such as those in IoT networks.

More Computing power Another significant advantage of DFL is that it provides more computing power for machine learning tasks [29]. From the developer's perspective, DFL is a zero-cost infrastructure because there is no need to maintain a centralized server.

Instead, all nodes can act as aggregation nodes and the computing power required to train the network is provided by the participating devices in cooperation.

5.4 Attacks in CFL

In this section, we will primarily discuss adversarial attacks in CFL, rather than malware attacks, as the latter can occur in many other machine learning applications and does not have much uniqueness in federated learning. Adversarial attacks in CFL, on the other hand, are of great research interest, and there have been numerous studies that classify these attacks, such as 1) taxonomy based on attack objective: to obtain data privacy and to reduce model performance [59]; 2) taxonomy based on attack strategy: passive attacks and active attacks [60]; 3) taxonomy based on phase: training phase and inference phase [61]. In this article, however, we will use the following taxonomy based on attack location: client-side, server-side, and federation-side attacks. The reason for this is that this classification allows us to more clearly identify the connections and differences between these attacks in CFL and DFL (the next section). By doing so, we'd like to provide a more comprehensive understanding of the adversarial attacks in federated learning and their impact on the privacy and security of the system.

5.4.1 Client-side attacks in CFL

Although CFL does not require centralized entities to maintain the security and privacy of collected data as in the data-centered approaches, the data is still distributed among local participants, providing potential targets for attackers. Moreover, as local participants need to compute model updates on their own and keep a copy of the latest global model, this introduces new attack surfaces for adversaries. Therefore, the significant threat surface in the CFL system is the set of local participants who are computing the model updates with their own data.

5.4.1.1 Poisoning Attacks

In poisoning attacks, attackers are assumed to be able to manipulate one or multiple participants to upload malicious updates to the server. These attackers can manipulate existing legitimate participants or can also directly join the federated network from the outside. To carry out poisoning attacks, attackers can employ mainly two methods [61]: 1) data poisoning attacks that disrupt the integrity of local data; 2) model poisoning attacks that directly manipulate the local training process to generate malicious model updates.

A. Data Poisoning

Data poisoning attacks indirectly compromise global model integrity by inducing malicious model updates through the use of compromised data. The objective of the attacker could be to reduce the performance of the global model or to insert a backdoor into it.

Label-flipping: Label-flipping involves an attacker intentionally mislabeling a portion of the training data in order to introduce errors into the model [62]. This technique can be used to target specific classes, and the ultimate goal is to reduce the accuracy of the global model. For example, an attacker might flip the labels of images of cats to make the model incorrectly classify them as dogs.

Backdoor data poisoning: Backdoor Data Poisoning is a type of data poisoning attack with the objective of manipulating the output of the model in a targeted manner. This is achieved by modifying a small subset of features or by adding small watermarks 5.2 to the original training dataset [63]. The inserted backdoors will then act on specific triggers in the inputs, causing the model to produce the desired outputs. Notably, the accuracy of the poisoned model remains unaffected, allowing the attacker to maintain the appearance of a well-functioning model while exploiting it for their own purposes. This type of attack is particularly insidious as it can evade detection by traditional model validation methods, making it difficult to detect and mitigate.

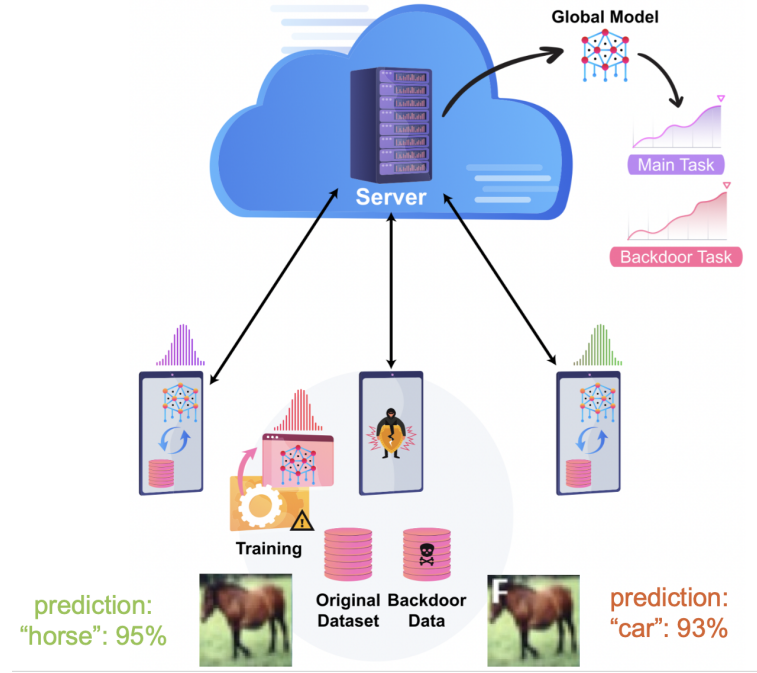


Figure 5.2: Attackers add watermarks into the dataset as a backdoor in the model, then the model predicts "horse" as "car" with high confidence.

B. Model Poisoning

Model poisoning attacks directly manipulate the local model updates or the learning rule [59], and like data poisoning attacks, they can also be classified into untargeted and targeted attacks.

Model updates manipulation: The objective of an untargeted model poisoning attack is to corrupt the global model to reduce the prediction accuracy of the model. In general, this type of poisoning attack can also be considered as an optimization problem. [10] proposed an optimization-based model poisoning attack and injecting adversarial neurons

in the redundant space of a neural network. Thus, an attacker can poison the model updates without changing the local data. Targeted model poisoning attacks [12] are very similar to targeted data poisoning attacks in that they use backdoor data to train backdoor model updates, and thus embed triggers to the model. Furthermore, to amplify the impact of the model updates, the attacker can decompose the trigger into separate local patterns and embed them into the local training data of different malicious clients using Distributed Backdoor Attack (DBA) [11].

Training rule manipulation: Attackers can also manipulate the training rules of a local computation to compromise the global model. This can be achieved through various techniques such as adding a penalty term to the objective function or changing the learning rate of the local computation. Such attacks can have a significant impact on the performance of the model, as the changes to the learning rules can alter the way the model learns and adapts to the data. Adding a penalty term to the objective function can bias the model toward certain features or limit its ability to learn certain patterns while changing the learning rate can affect the convergence rate and stability of the model. The authors of [12] have reported an instance of advanced training rule manipulation, in which they successfully executed a stealthy targeted model poisoning attack. The authors accomplished this by incorporating a penalty term into the objective function, aiming to minimize the distance between the weight update distributions of both malicious and benign samples.

5.4.1.2 Free-riding Attacks

Free-riding attacks [65, 64] in CFL can be perceived as a variation of poisoning attacks, though their intent is not malicious but rather selfish. These attacks typically originate from participants who aim to reap the benefits of the federation service without actively participating in the training process. The reasons for this could vary, such as a lack of adequate data, computing resources, or even concerns surrounding privacy.

In order to simulate participation in CFL, these selfish participants may generate fake updates randomly or use previous model versions to deceive the system [9]. However, such actions can still hinder the overall performance of the FL model. While the degree of damage caused by free-riding attacks may not be as severe as that of poisoning attacks, it still poses a serious threat to the integrity of the global model.

5.4.1.3 Evasion Attacks

Evasion attacks [66, 67] pose a significant threat to the accuracy and integrity of deployed models in local participants. The primary objective of such attacks is to manipulate the model into producing incorrect outputs when attempting to make predictions. This can be achieved by carefully crafting and manipulating the data samples that are fed into the model during the inference stage. One popular method of conducting evasion attacks is through the use of "adversarial samples" [Fig 5.3]. These are essentially modified versions of original test samples that appear visually similar to the human eye but contain subtle changes that can confuse the deployed model, leading it to make incorrect predictions.

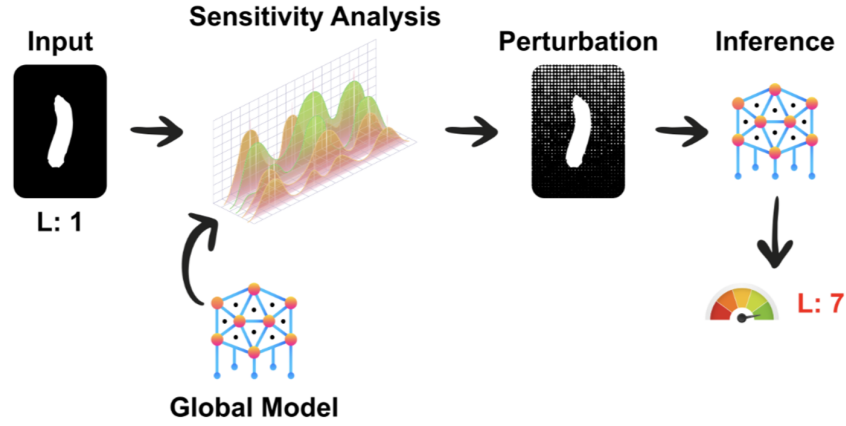


Figure 5.3: Attackers evade a model using adversarial samples.

The effects of evasion attacks can be severe, as they can significantly compromise the accuracy of a well-trained machine learning model. In practice, this renders the model unreliable for practical use, as the outputs generated by the model cannot be trusted.

5.4.2 Server-side attacks in CFL

CFL systems are vulnerable to various types of server-side attacks that can compromise the security and integrity of the system. Distributed Denial of Service (DDoS) attacks [68] on server and compromised or malicious servers are common types of server-side attacks that can cause significant damage to the system. Note that the devastation to the entire CFL would be enormous if the central server is unavailable or untrustworthy because CFL relies on the central server to coordinate.

DDoS attacks are launched by attackers or botnets to flood the server with a high volume of traffic, rendering it unavailable to legitimate users. Such attacks can severely disrupt the CFL system, causing significant delays or even a complete halt to the learning process. The compromise of the server itself is another type of server-side attack in CFL. This occurs when an attacker gains unauthorized access to the server by exploiting a vulnerability or using weak credentials. Once an attacker gains access, they can modify the learning process, extract sensitive data, or inject malicious code into the system. Malicious servers can also pose a threat to CFL systems. These are servers intentionally set up to steal data, disrupt the learning process, or manipulate the results.

5.4.3 Federation-side attacks in CFL

In addition to the previously mentioned client-side and server-side attacks, there are also attacks that target the communication process, which involves the transfer of model updates and global models across the federation network. Although CFL does not allow clients to upload raw data directly to the central server, there is still a significant volume of communications between them and a large amount of model information. As a result,

attackers can take advantage of this aspect of CFL and exploit the information that is being exchanged. This not only increases the attack surface but also puts the privacy of the participants at risk. Attackers can carry out a range of attacks, including eavesdropping, intercepting, and replacing models, which can lead to the inference of further private data of the participants. Man-in-the-middle attacks and inference attacks are two of the most representative types of attacks used to exploit this vulnerability. These attacks are not unique to CFL and are often discussed in other machine learning paradigms. However, in this section, it is important to examine how these attacks are applied in the CFL setting and how they are interconnected.

5.4.3.1 Man-in-the-middle Attacks

Man-in-the-middle attacks [69, 70] are a type of attack where an attacker tries to intercept the communication between two nodes and masquerades as the intended recipient to manipulate or eavesdrop on the communication. In CFL, the attacker could act as an intermediary between the clients and the central server to intercept the model updates from local participants and the global model from the central server [Fig 5.4]. By doing so, the attacker can alter the model updates and even inject malicious updates into the communication, leading to the compromise of the integrity and confidentiality of the model. Additionally, the attacker may also try to reduce the performance of the model by delaying the model updates, leading to slower convergence or even non-convergence of the model.

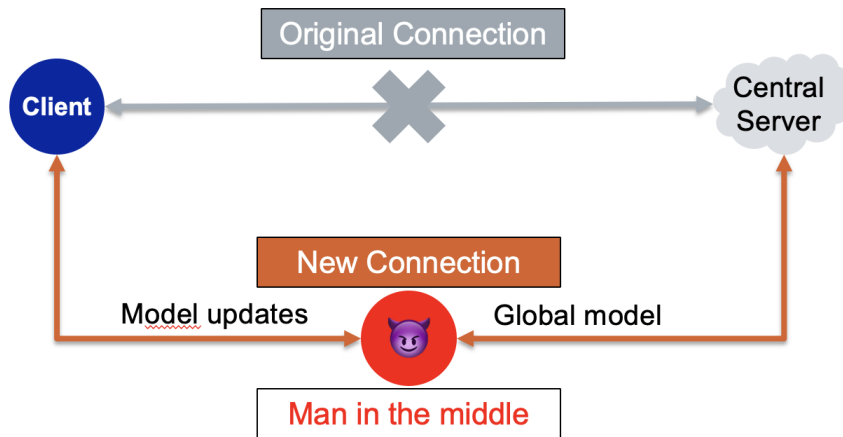


Figure 5.4: Man in the middle intercepts the model updates and global model.

The man-in-the-middle attack can also be used as a stepping stone for performing inference attacks. By intercepting and analyzing the model updates, the attacker may be able to infer sensitive information about the clients' data. For instance, by analyzing the model updates of a certain client over multiple communication rounds, the attacker may be able to infer their preferences or behavior, which could lead to privacy violations.

5.4.3.2 Inference Attacks

Inference attacks aim to extract sensitive data from the model updates and the global model in the federation network. These attacks assume that the attacker has at least black-box access to the model [17], which can be obtained through previously mentioned attacks, such as pretending to be a participant or intercepting connections. Inference attacks can potentially reveal confidential information about local participants, such as medical records, financial information, and personal preferences. These attacks pose a significant threat to the privacy and security of the federated learning system.

Membership Inference Attacks Membership inference attacks exploit the overfitting nature of machine learning models to infer the membership of a sample in the training dataset. The attackers aim to determine if a specific data point was used in the training of a model. To achieve this, the attacker trains multiple “shadow models” with similar architecture and training data as the target model to imitate its behavior, but with a different set of samples. [17] developed several methods to generate training data for the shadow models, including using black-box access to the target model, statistics about the population, and potentially noisy versions of the target’s training dataset. Based on the confidence scores of the outputs, the attacker can infer whether a specific input was present in the training data or not. If the confidence score for a particular input is high, hence it was likely present in the training data [Fig 5.5]. This way, the attacker can determine the membership of a sample with high accuracy. Member inference attacks can be easily performed in CFL if there is a lack of protection for the models. They can particularly damage CFL scenarios involving sensitive data, such as medical records or financial information. An attacker could use these attacks to determine if a specific individual was part of the training data, thereby breaching their privacy.

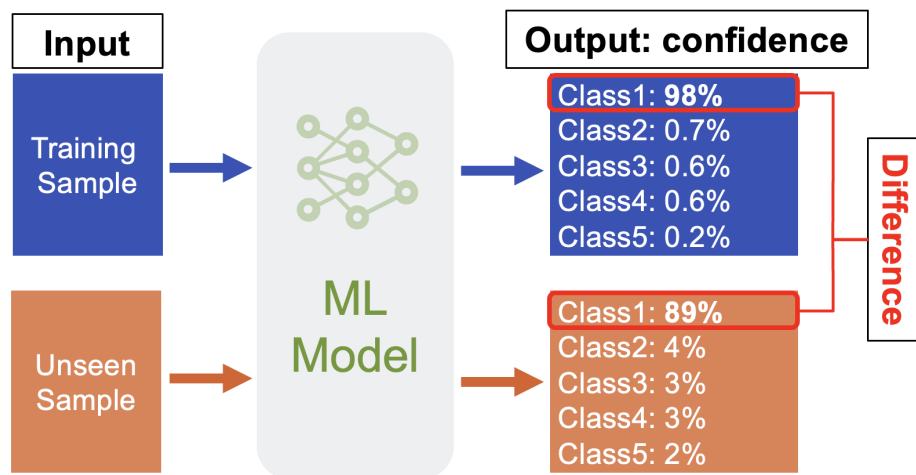


Figure 5.5: Attackers can analyze the output confidence of shadow models to infer whether a given sample has been used in the training dataset.

Model Inversion Attacks In model inversion attacks, the attacker’s goal is to reconstruct sensitive data by inverting the target model. To achieve this, the attacker first generates a large number of input and output pairs from the target model in federated learning. For example, [71] showed a case of a facial recognition model, the input might be a face image, and the output might be a list of information about the person, such as their name, age, and address. The attacker then switches these pairs and trains an inversion model on the resulting dataset. The inversion model learns to map the output back to the input, effectively inverting the target model’s behavior. To create more powerful reconstructions, the attacker may use a Generative Adversarial Network (GAN) [72] as the inversion model. GANs are a type of neural network that can generate highly realistic inputs by learning to mimic a given dataset. By using a GAN as the inversion model, the attacker can create highly accurate reconstructions of sensitive data. Once the inversion model is trained well, the attacker can feed a list of information into it, and the model will output an inferred input (e.g., a face image) [Fig 5.6]. The inferred input and the victim input can be almost indistinguishable in the case of [71], meaning that the private data of some local participants has been leaked. [18] investigated the possibility of performing gradient inversion for image classification tasks in the context of a federated network. Specifically, if the updates exchanged among the participants in the federated network are in the form of gradients, then it is feasible to launch a gradient inversion attack using these leaked gradients, without requiring black-box access to the model.

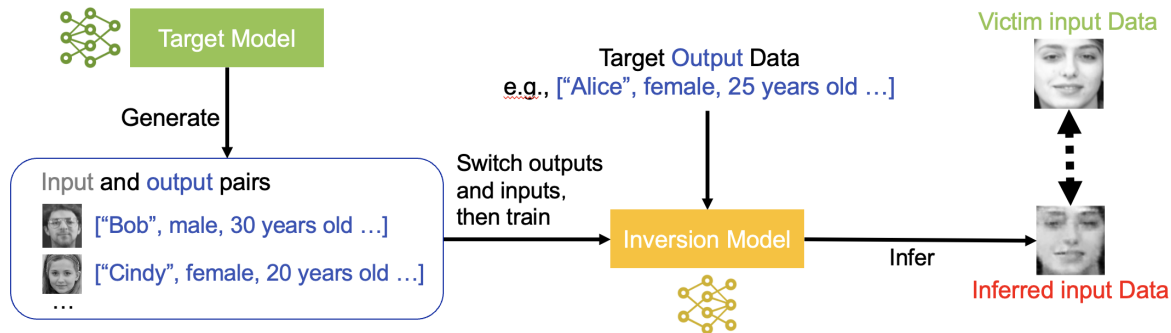


Figure 5.6: Attackers generate the input and output pairs from the target model to train an inversion model.

5.5 Attacks in DFL

Most of the existing literature in the field of Federated Learning has been focused on the study of attacks that occur in CFL [29]. Regrettably, attacks in DFL have not received sufficient scrutiny. To address the crucial question of how attacks in CFL impact DFL as well, and to identify the connections and differences between them, this section will provide a more detailed discussion of attacks in DFL based on the previous analysis of attacks in CFL. Given that CFL and DFL are both subclasses of FL, they unavoidably share many similarities when confronted with attacks. In line with our categorization of CFL attacks in the previous section, client-side and federation-side attacks remain effective in DFL

as well, whereas DFL prevents server-side attacks by design. Nonetheless, the serverless environment of DFL is the fundamental discrepancy between it and CFL, which also brings about new vulnerabilities. Certain attacks that were originally applied in peer-to-peer network topologies can also jeopardize DFL systems, including Sybil attacks [76] and Eclipse attacks [35].

5.5.1 Connections between attacks in CFL and DFL

CFL can be viewed as a special case of DFL in which the network topology is star-shaped. As a result, CFL and DFL share many similarities and are susceptible to many of the same attacks. [Table 3] shows that most attack types will also pose a threat in DFL, albeit with a slightly different attack surface. Specifically, poisoning attacks, evasion attacks, free-riding attacks, DDoS attacks, compromised or malicious server attacks, man-in-the-middle Attacks, and inference attacks are all relevant attack types for both CFL and DFL. However, DFL does not have any attack surface related to the central server, which eliminates the compromised or malicious server problem that is present in CFL.

Table 5.3: Comparison of attacks in CFL and DFL

Attack Type	Attack surfaces in CFL	Attack surfaces in DFL
Poisoning Attacks	On certain clients(data & model updates)	On certain clients(data & model updates)
Evasion Attacks	On certain clients(prediction)	On certain clients(prediction)
Free-riding Attacks	On certain clients(model updates)	On certain clients(model updates)
DDoS Attacks	On server and certain clients	On certain clients
Compromised or Malicious Server	On server	(Solved it essentially)
Man-in-the-middle Attacks	Communication between clients and server	Communication among clients
Inference Attacks	On communication and model	On communication and model

Both CFL and DFL are susceptible to client-side attacks that can adversely affect the training and prediction process and degrade the performance of the model. The attack surfaces in both types of federated learning are almost identical, with poisoning and free-riding attacks being examples of attacks that indirectly manipulate local data or directly manipulate model updates to generate malicious or useless updates before they are uploaded by local participants. Evasion attacks, on the other hand, cause incorrect predictions of the local models after they have been deployed to local participants. It is worth noting that these client-side attacks are not significantly impacted by the network topologies of CFL and DFL, as they occur either before transmitting model updates or after deploying the model. However, DFL differs in the aggregation phase from CFL in

the face of malicious model updates, and we will specifically analyze the weaknesses of DFL later.

In addition, man-in-the-middle attacks occur in communication between clients and the server in CFL, while in DFL, they occur among clients due to the absence of a centralized server. The lack of a centralized server in DFL makes it less susceptible to attacks on the server itself but introduces new challenges in terms of protecting communication and coordination among clients. In order to carry out Inference attacks, the attacker must have at least black-box access to the model [17], regardless of whether it is in the context of DFL or CFL. The precondition for achieving such access can be satisfied by pretending to participate and intercepting communications. Once the attacker has gained access to the target model, the subsequent processes of training the shadow models or inversion models become irrelevant to whether it is in a CFL or DFL environment.

Another important observation is that DFL has no attack surface on the server, which means that DFL does not face a situation where the entire system will be completely stopped once the server is compromised by a DDoS attack. Furthermore, DFL essentially solves the problem of a compromised or malicious server. This property is completely determined by the design of DFL itself and serves as the main motivation for studying DFL on top of CFL.

5.5.2 Vulnerabilities of DFL Facing Attacks

Although DFL has many benefits in a serverless setting, such as not requiring a trusted central server and being more resilient to network attacks such as DDoS attacks on servers, it also introduces several new weaknesses.

Firstly, in DFL, participants communicate directly with their neighbor nodes, rather than using a central server as an intermediary [73]. This results in the number of connections in a DFL network, depending on the network typology, often being several orders of magnitude higher than in a CFL. This high number of connections creates an opportunity for highly propagated malware to take advantage of and spread rapidly throughout the network.

Secondly, DFL lacks a powerful central server to coordinate the learning process of the federation, especially when the number of participants in the federation grows exponentially [29]. Although ideally, participants in the network can self-manage, in real-world application scenarios, the connections between participants are intermittent and often weak [39]. This poses a challenge to the self-management of DFL, making the decentralized scenario more vulnerable to certain attacks than the CFL scenario.

Thirdly, each participant in DFL can act as an aggregator instead of delegating a central server to accomplish aggregation as in CFL [29]. The process of aggregating model updates is a key aspect of federation learning, and if controlled by a large number of malicious nodes, it would be highly disruptive to the overall DFL.

Finally, DFL also inherits some of the attacks faced by inherently distributed networks, such as the famous eclipse attack on the Bitcoin network. Such attacks pose a greater threat to DFL.

5.5.2.1 Vulnerable to Malware Attacks

Due to the lack of a central server for coordination, DFL is more vulnerable to malware attacks as compared to CFL. Participants in DFL communicate directly with each other, and the absence of a central server can make it challenging to detect malware and prevent it from causing damage to the network. In networks with a large number of participants, this vulnerability is even more pronounced.

Additionally, the number of connected edges in DFL can be much larger than in CFL, especially in randomly connected and fully connected network topologies 5.4, making them more susceptible to security risks. Network participants usually exchange parameters of the local model for contributing to DFL with a high frequency. In this sense, the exchanged messages are obtained by neighboring participants, extracted, and processed for aggregation or forwarding in the case of proxy participants [29]. However, the information obtained may have been modified during the exchange, adding malicious data using rootkits, or the inclusion of the participant in a botnet. In this way, the node would be infected and could spread easily and quickly among the participants of the decentralized federation, increasing its expansion when there are many connections between neighbors.

Table 5.4: Number of connections in different network topologies

Network Typology	Number of connections for each node	Number of all connections in network
Star structure (CFL)	1	N
Ring Structure (DFL)	2	N
Random (DFL)	M (average)	$MN/2$
Fully connected (DFL)	$N - 1$	$N * (N - 1) / 2$

5.5.2.2 Vulnerable to Poisoning and Free-riding Attacks

The poisoning and free-riding attacks have been discussed in previous sections, and both of them produce malicious model updates that can cause significant harm to the performance of the overall model. The decentralized approach, while reducing the need for a central server, makes it more challenging to detect malicious updates since there is no single point of aggregation.

Although each node in a DFL can act as an aggregator, the process of aggregation is distributed among all the nodes, and they are typically not as powerful as a central server to implement sufficiently sophisticated detection and aggregation algorithms. In a CFL, the central server can play an active role in monitoring the model updates and detecting

any malicious activities. However, in a DFL, the lack of a powerful central server to coordinate the learning process makes it more vulnerable to these attacks.

5.5.2.3 Vulnerable to Sybil Attacks

Sybil attacks are a significant security threat in decentralized federated learning, where an attacker creates multiple fake pseudonym nodes to flood the network and gain control over the DFL process, with the aim of reducing the model performance. In a decentralized network, there is no central authority to verify the identity of nodes, making it difficult to distinguish between pseudonym and non-pseudonym nodes in DFL settings [29]. The attacker can exploit the lack of a central authority to control the DFL process by introducing a large number of fake nodes, which can communicate with each other and share false model updates, leading to a degradation in the overall model performance.

These fake nodes can also be used by the attacker to act as aggregators, enabling them to exert more control over the model updates [Fig 5.7]. By controlling the aggregation process, the attacker can easily inject malicious data, and selectively forward, drop and suppress model updates from legitimate nodes, thereby compromising the performance of the overall model.

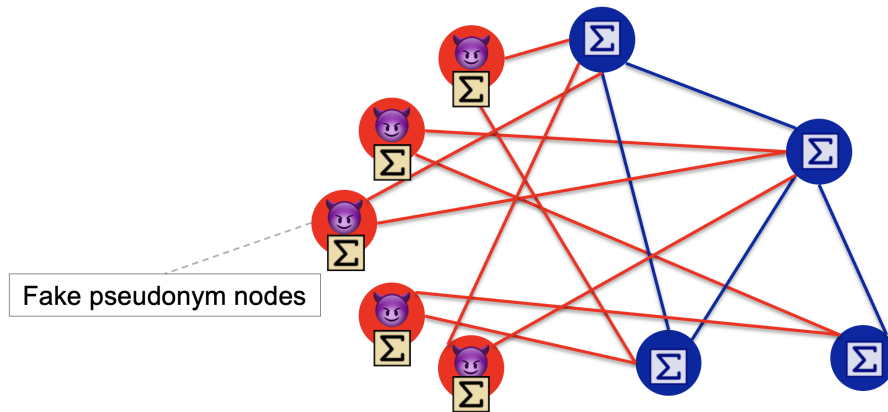


Figure 5.7: Sybil attacks create multiple fake pseudonym nodes to flood the network and these nodes can act as aggregators as well.

5.5.2.4 Vulnerable to Eclipse Attacks

Eclipse attacks in the federation can compromise the communication channels of target nodes, isolate them from the rest of the network, and manipulate the information they receive. These attacks are particularly concerning in DFL networks because they lack a central authority to manage and monitor communication channels. Instead, the nodes in DFL networks communicate with each other in a decentralized manner, making it easier for attackers to manipulate the network connections and gain control over the information received by victim nodes.

To understand the threat of Eclipse Attacks in DFL, it is crucial to comprehend their mechanism. Eclipse attacks were initially proposed to lower the security threshold of the Bitcoin network by partitioning it into sub-networks [35]. In this scenario, attacker nodes control more than 51% of the mining power in each sub-network, enabling them to perform a 51% attack with less than 50% of the mining power. This concept can be extended to DFL networks [74], where the ratio of attacker nodes in each partitioned sub-network can be higher than the overall ratio in the entire network due to the distributed nature of the network [Fig 5.8].

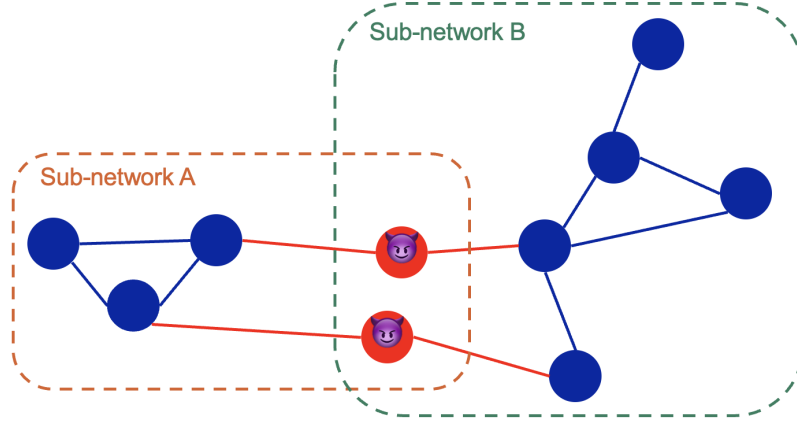


Figure 5.8: By eclipsing and partitioning the distributed federation network, malicious nodes gain more control ratio with less effort.

Moreover, depending on the topology of the distributed network, an attacker may isolate more participants with less effort. Unlike CFL, where an attacker can only isolate exactly one victim per cutting-off operation, they would spend more time manipulating the routers and switches in the client-server network. Therefore, the decentralized nature of DFL networks, coupled with the possibility of high control ratio in each partitioned sub-network and the potential ease of isolating victims, make them more vulnerable to Eclipse attacks.

5.6 Defense Methods in CFL

To mitigate the potential risks of attacks in centralized federated learning, various defense methods have been proposed. These techniques aim to protect the models and data used in collaborative learning while maintaining their integrity and privacy. The effectiveness of various defense strategies against various attackers will be covered in this section. We will specifically look at things like each defense strategy's focused phase, the primary attacks it can mitigate, the cost of implementation, and how it affects the performance of the learning process.

To categorize those defense approaches, many papers classified them according to the attacks they defend against like in [6], Sikandar introduced poisoning attacks, Inference attacks, Backdoor attacks, etc., and defense mechanisms accordingly. And in [5], Gosselin

provided with different taxonomy of defense, and two basic defense strategies were introduced: (i) the proactive one, which is upstream before the assault and seeks to defend against attacks; and (ii) the reactive one, which is used after the attack occurred and aims to mitigate the effects and make a patch. Also in [9], Bouacida states proactive and reactive defense methods as well. Proactive defense is a cost-effective way to figure out the threats and related risks prematurely. The reactive defense works by identifying the attack and taking precautionary measures. Fig 5.9 illustrates the FL defenses that have been mentioned and are utilized to ensure robustness against adversarial attacks.

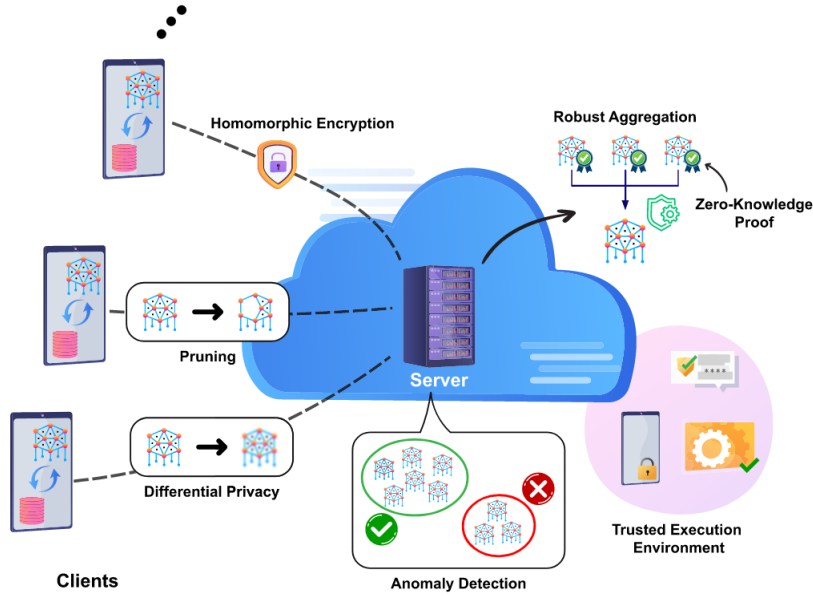


Figure 5.9: Most popular defense mechanisms to ensure FL security

In addition, [7] introduced a multi-phase framework of FL including data and behavior auditing, model training and model predicting. Separately, Liu illustrates attacks and defense methods accordingly in each phase. In the data and behavior auditing phase, adversaries can cause damage through social engineering, penetration attacks, backdoor attacks, and advanced persistent threat attacks, etc. By auditing the data quality of the local workers and evaluating behaviors, the untrusted local workers can be eliminated. In the training phase, FL mainly faced poisoning attacks and privacy inference attacks which are both white-box attacks. Existing strategies to mitigate private inference are usually based on the processing of shared gradient information, including Compression Gradients, Cryptology Gradients and Perturbation Gradients. In the model predicting phase, there are still security and privacy threats faced evasion attacks, Model Inversion and Membership Inference as well. To defend against those attacks, Empirical defense, Certified defense, model structure defense and information obfuscation defense was introduced.

In this review, we introduce a new taxonomy of defense approaches. These methods can be broadly classified into the following categories shown in Table 5.5: Cryptographic approaches, including Differential Privacy, Homomorphic Encryption and Secure Multiparty Computation. These techniques are based on encryption to protect data privacy and security. Aggregation algorithms: including Secure Aggregation and Robust Aggregation. These algorithms are mainly used for confidentiality and robustness when aggregating

models. Client Identification: including Recognizing Legitimate Clients which is used to identify legitimate clients and prevent illegal clients from compromising the federation learning system. And Model Training: including Federated Distillation, Adversarial Training and Federated Multi-Task Learning which are mainly used to improve the efficiency and robustness of model training.

Table 5.5: Taxonomy of defense approaches

Taxonomy	Approaches	Description
Cryptographic approaches	Differential Privacy	Based on encryption to protect data privacy and security
	Homomorphic Encryption	
	Secure Multiparty Computation	
Aggregation Algorithms	Secure Aggregation	Confidentiality and robustness when aggregating models
	Robust Aggregation	
Client Identification	Recognizing Legitimate Clients	Identify legitimate clients and prevent illegal clients from compromising the federation learning system
Model Training	Federated Distillation	Improve the efficiency and robustness of model training
	Adversarial Training	
	Federated Multi-Task Learning	

5.6.1 Cryptographic approaches

In this section, we will introduce Differential Privacy (DP), Homomorphic Encryption (HE), and Secure Multi-Party Computation (MPC), three cryptographic methods that are particularly relevant to secure centralized federated learning (CFL). We are aware that other additional cryptographic methods, such as functional encryption, verifiable computation, identity-based encryption, and attribute-based encryption, can be utilized to protect CFL as well. We nevertheless concentrate on the methods we consider to be the most effective and relevant for mitigating attacks in CFL.

A. Differential Privacy

To prevent private information from being divulged to some extent by analyzing the differences in parameters trained and uploaded by the clients, differential privacy is introduced to prevent information leakage by adding artificial noise [13]. The fundamental idea behind this is to add noise to the output of computation in such a way that it makes it difficult for an attacker to determine the presence or absence of a particular individual in the dataset. The amount of noise added is controlled by a parameter called the privacy budget, which determines the trade-off between privacy and accuracy. To maintain

the privacy of the data owned by individual nodes and model parameters generated in each client, centralized federated learning can use DP to mitigate attacks. When machine learning training is done centrally, the model is often maintained by the server. The gradients computed by each client and pooled by the server to update the model can be added noise via differential privacy. This makes it possible for the server to still carry out aggregation while preserving the privacy of the data held by specific clients [14].

One common approach to implementing differential privacy in centralized federated learning is to use the concept of gradient perturbation. This method involves perturbing the gradients calculated by each client with a small amount of noise that is precisely calibrated to make sure the resulting model update complies with a specified privacy guarantee. The model is then updated by the server aggregating the perturbed gradients [15].

Overall, DP is a crucial method for preserving the confidentiality of specific client information in centralized federated learning. It can be used to give solid privacy guarantees while still enabling machine learning training on the server. To make sure that the resulting model update satisfies the desired privacy guarantee, it is essential to properly calibrate the quantity of noise supplied, because, like many privacy-enhancing techniques, DP has trade-offs. The trade-off between privacy and data utility is one of them. The precision of the data will most likely be decreased by adding noise, which makes it more challenging to derive insightful conclusions. In order to obtain both strong privacy protections and adequate data utility for the specified use case, the quantity of noise supplied must be carefully weighed according to specific needs.

In some circumstances, if model accuracy is more important, privacy protections may have to be given up in order to preserve reliability and accuracy. The price of applying DP computationally is another trade-off. When applying DP, complicated mathematical operations are frequently used, including the creation of random numbers and the use of differential privacy techniques. The speed and scalability of the training may be constrained by the computationally expensive nature of these operations. Therefore, while employing DP, it is necessary to carefully weigh the trade-off between privacy and computational expense.

B. Homomorphic Encryption

In the aggregation of models in the central server, the aggregator can see the unencrypted local and aggregated models. To mitigate that, we introduce homomorphic encryption features to CFL in order to further strengthen the nodes' data privacy and security in scenarios where the aggregator runs in a less reliable environment and the nodes seek to prevent exposing the local models and the aggregated models to the aggregator [16]. Homomorphic encryption (HE) is a form of encryption that enables performing computations on the encrypted data without decrypting it. The computation's results are still in encrypted form, and when they are decrypted, they yield the same output as if the computations had been done on the unencrypted data.

Before uploading their local model updates to the aggregator in federated learning, the nodes can homomorphically encrypt them which is illustrated in Fig 5.10. The aggregator only sees the homomorphically encrypted local model updates, therefore it is unable to draw any conclusions from this data. In particular, the aggregator is unable to extract data

from local training data by reverse engineering the local model updates. The aggregator combines the encrypted local model updates to produce an encrypted aggregated model. The nodes receive the encrypted aggregated model from the aggregator, which they decode and use to begin the subsequent training cycle.

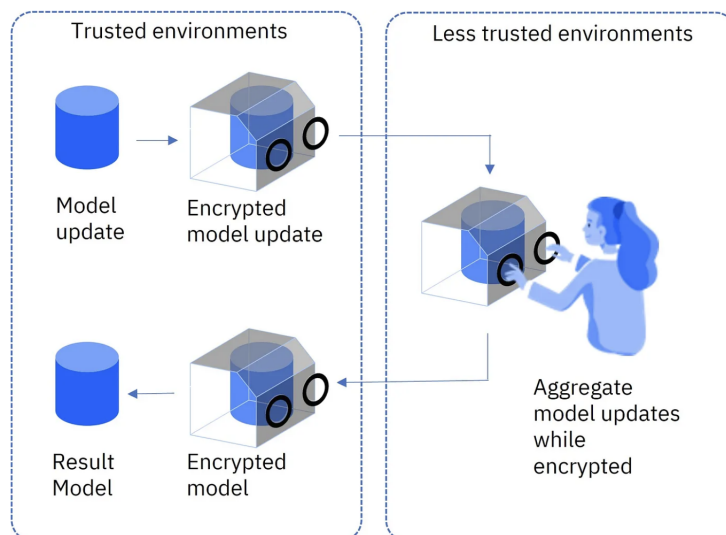


Figure 5.10: Most popular defense mechanisms to ensure FL security

Homomorphic Encryption can mitigate various different kinds of attacks, such as: Insider threats: HE can reduce the possibility of insider threats involving unauthorized access to sensitive data by allowing data to be handled without first being decrypted. Man-in-the-middle attacks: By encrypting data in transit and confirming the accuracy of the data at the receiving end, homomorphic encryption can guard against man-in-the-middle attacks. Also, attacks that attempt to steal or alter models, known as "model extraction attacks," can be mitigated by encrypting the parameters of the model, making it more difficult for attackers to do so. It is important to note that while HE can mitigate these types of attacks, it also comes with a trade-off in terms of computational overhead and increased processing time. As such, it may not be suitable for all applications and scenarios.

C. Secure Multiparty Computation

According to the previous discussion, by introducing noise to the data, intermediary calculations, or outputs, differential privacy can aid in preventing such reverse engineering. Even after reconstruction, the input data set will not contain any user's precise data. Formally speaking, differential privacy is a mathematical idea that, by varying values, ensures statistical indistinguishability for individual inputs. It is widely explored in the literature how DP machine learning algorithms can be used in centralized federated learning. Major corporations have embraced the technology; for instance, Apple utilizes it for web search auto-completion. DP is used to add a layer of randomness so that attackers even with more information still don't know the raw data. However, there is a trade-off: the addition of randomness to the data collection compromises accuracy while protecting user privacy.

In order to mitigate that, Secure multi-party computing (MPC) is introduced to the aforementioned issue that ensures anonymity without sacrificing accuracy. Without disclosing

their private inputs to one another, multiple parties can work together to calculate a common function of interest using Secure MPC. If the parties only learn the outcome, and no other information, an MPC protocol is considered secure [20]. For instance, in a practical situation, a group of workers might want to figure out their average pay without any of them disclosing their particular pay to any other workers. MPC can be used to do this work by allowing the average income to be the only piece of information that is revealed.

Therefore, we can achieve secure federated learning by combining methods like DP and MPC, which can be done as follows: Nodes train a model locally on their own data and then send the weights (or parameters) of their model to the server using an encryption scheme that still ensure the server to process the encrypted data. Specifically, the server can compute a weighted average of all the encrypted weights received from users, but it cannot discover the original weights or raw data of any nodes.

5.6.2 Aggregation algorithms

For federated learning (FL) systems to be secure and private, aggregation methods are essential. These algorithms are in charge of gathering the model updates obtained from the involved customers while protecting the confidentiality of their information. These techniques could, however, be the targets of attacks like model poisoning and data inference assaults if there are malicious clients present. Several aggregation strategies have been put out in the literature to mitigate the impact of these attacks.

A. Secure Aggregation

Secure Aggregation is a class of Secure Multi-Party Computation algorithms wherein a group of mutually distrustful parties $u \in U$ each hold a private value and collaborate to compute an aggregate value without revealing to one another any information about their private value except what is learnable from the aggregate value itself [21]. In CFL, the leakage of information can be caused by the updates in each iteration and it is possible to obtain the private training data from the publicly shared gradients[23]. So, how can the leakage brought on by upgrades be minimized? The solution is to use secure aggregation, a technique that only shows the coordinator the aggregated result after covertly adding up all the changes. The coordinator can seldom learn anything significant about a single participant because the outcome is formed by all the participants.

At first glance, Secure Aggregation is quite similar to cryptographic techniques like Homomorphic Encryption, Differential privacy and MPC in practice. Indeed, secure aggregation could be initialized by those cryptographic methods, but it just makes the aggregation step of FL secure and does not ensure a fully secure FL procedure [22]. The connection between them is that MPC offers a significantly better level of security at the expense of costly cryptographic operations, which frequently raises the cost of computation and communication. The security requirements are loosened by secure aggregation, allowing for a clearer and more effective implementation. It's important to note that MPC is already quite effective for simple models with few participants. However, CFL with secure aggregation is probably a better workable approach for challenging jobs like training on millions of mobile phones.

B. Robust Aggregation

In CFL, it relies on the aggregation of updates contributed by participating devices where the aggregation is privacy-preserving but sensitive to corrupted updates. This can happen when adversaries intend to attack the system or due to failures in low-cost hardware. In CFL, the typical arithmetic mean aggregation is not corruption-resistant, as even one corrupted update during a round is enough to impair the overall model across all devices. So, in [24], Pillutla designed a novel robust aggregation oracle based on the classical geometric median and analyzed the convergence of the resulting federated learning algorithm, RFA, for least-squares estimation and show that the proposed method is robust to update corruption in up to half the devices in federated learning with bounded heterogeneity.

Similar to the non-robust secure average oracle it builds upon, the robust aggregation oracle protects privacy. They demonstrate its convergence for additive model least squares estimation and provide experimental findings for three tasks in computer vision and natural language processing using linear models and deep networks. The robust aggregation strategy performs better than the traditional aggregation approach in terms of robustness when the amount of corruption is high while remaining competitive in the regime of low corruption and is independent of the level of corruption. Note that Secure Aggregation and Robust Aggregation can be used together to improve the security and robustness of the CFL model.

5.6.3 Client Identification

Client identification methods in CFL refer to techniques used to verify the authenticity of clients participating in the FL process. These methods are critical to ensure that only legitimate clients are allowed to participate in the FL process, which can prevent malicious attacks and ensure the security and privacy of the system.

A. Recognizing Legitimate Clients

A critical step in ensuring the security and privacy of the federated learning system is to identify authentic clients in CFL. The purpose of client identification is to distinguish between genuine clients who take part in training and malevolent clients who wish to obstruct instruction or steal confidential data. Utilizing authentication measures to guarantee that only authorized clients can take part in the training process is one method for client identification. Clients can be made to submit legitimate login information or certificates in order to access the system. This strategy, meanwhile, has drawbacks because compromised credentials or certificates can be used by attackers to pose as trustworthy clients.

An alternative strategy is to employ anomaly detection techniques to spot trainees who act in an unusual way. A client may be malevolent if, for instance, it sends noticeably more data than other clients or if the statistical distribution of the data it sends differs noticeably from that of other clients. Techniques like clustering, principal component analysis, and support vector machines can be used for anomaly detection. Recent studies [25] have also concentrated on employing machine learning models to recognize malicious

clients. These models are trained on features such as the amount of data sent by each client, the statistical properties of the data, and the timing and frequency of the client's interactions with the system. The models can then predict whether a client is legitimate or malicious based on these features.

Overall, identifying genuine clients in CFL is a difficult task that requires the use of authentication measures, anomaly detection, and machine learning approaches. The security and privacy of the federated learning system can be improved by successfully recognizing and deleting bad clients.

5.6.4 Model Training

A. Federated Distillation

Federated Learning becomes quite expensive when communication resources are few, especially for contemporary deep neural networks with a large number of model parameters. In this sense, federated distillation (FD) is a strong distributed learning approach that only trades model outputs with dimensions that are frequently considerably lower than the model sizes (for instance, 10 labels in the MNIST dataset) [26]. In FD, a large pre-trained model is used as a teacher model, and the goal is to distill the knowledge from the teacher model into a smaller student model that is trained on client data. The teacher model is kept private on the server and is not exposed to the clients. During training, the clients download the current student model, train it on their local data, and upload the updated model to the server. The server then aggregates the uploaded models to produce a new global model, which is used to update the student model for the next round of training. FD is quite effective against poisoning attacks because it is resistant to the influence of malicious updates. Even if a client sends a poisoned model update, it is unlikely to affect the final global model because the teacher model is used as a reference to ensure that the updates are consistent with the overall goal of the learning task. In addition, recent research has shown that federated distillation can be effective in defending against various types of poisoning attacks, including data poisoning, model poisoning, and backdoor attacks. By distilling knowledge from multiple local models, the global model can be made more resilient to the effects of poisoning attacks. In addition, federated distillation can also reduce communication overhead and improve the convergence speed of the global model. Adversarial Training Federated learning models can be made more resilient to adversarial attacks by using a technique called adversarial training. In conventional machine learning, the model is trained on a clean dataset with the expectation that the training data would always be representative of the real world. However, in adversarial environments, attackers can purposefully alter the input data to trick the model, resulting in false or harmful results. Adversarial examples are modified versions of the original input intended to lead to misclassification or other errors in the model, and they are added to the training data during adversarial training. The model can be trained on these adversarial samples to become more resistant to similar attacks in the future. When it comes to CFL, adversarial training can be used to improve the robustness of models trained on decentralized data. Each client can generate adversarial examples based on their local data, which are then aggregated and used to train the global

model. This approach can help defend against various attacks, including data poisoning and model poisoning attacks [27]. For example, in a data poisoning attack, an attacker can manipulate the data on one or more clients to bias the training of the global model. By using adversarial training, the model can learn to detect and mitigate such attacks by being trained on adversarial examples that simulate these scenarios. Similarly, in a model poisoning attack, an attacker can modify the model parameters on a subset of clients to cause misclassification or other errors in the global model. By training on adversarial examples that mimic these types of attacks, the model can learn to be more robust to such manipulations. Federated Multi-Task Learning Federated Multi-Task Learning (FML) is an approach in which multiple tasks are learned simultaneously across multiple clients in a federated learning setting. By utilizing the similarities between tasks and the diversity of data between clients, FML seeks to enhance the system's overall performance. FML can be used for a variety of machine-learning applications, including speech recognition, image classification, and natural language processing. FML can be utilized in CFL to enhance system performance while preserving data privacy as well. FML enables clients to jointly learn a number of related tasks using a shared model as opposed to training separate models for each task on each client. FML can enhance the accuracy and generalization of the model while lowering the risk of overfitting by taking advantage of the similarities between tasks and the variety of the data across customers. FML can mitigate various types of attacks, such as data poisoning attacks and model inversion attacks. In a data poisoning attack, a hostile client purposefully feeds erroneous data to the main server to reduce the model's performance. Since the shared model in FML is trained using data from several clients, it is challenging for a single rogue client to have a major negative impact on the model's performance. An attacker attempts to deduce sensitive information about the training data or the model itself in a model inversion attack. By making it more difficult for an attacker to deduce information about the data of particular clients, FML can strengthen the privacy of the model by jointly training numerous tasks on multiple clients.

5.7 Defense Methods in DFL

5.7.1 Defense methods inherited from CFL ecosystem

In DFL, countermeasures are also essential to protect from malicious nodes. Similar to how the CFL strategy protects privacy, the federated network created by the DFL technique relies on participants sharing model parameters to avoid sending raw data. However, the training data, updates, and model activation can be compromised through the gradients transferred during the distributed training process which can also happen in CFL as well [28]. So, defense measures in DFL can be inherited from CFL accordingly. For privacy-preserving, the majority of those countermeasures use standard Differential Privacy (DP) relaxations to inject less noise while obfuscating updates during decentralized federation at the expense of each local model's accuracy. The most restricted participants' performance can end up suffering as a result of this method, which would be undesirable for the federation. Since there are several communications between each node, the DFL technique is more challenging to adapt than CFL. For each set of model parameters exchanged

on each communication link, each participant needs to perform a significant amount of computation to add or reduce noise [29].

To avoid sample reconstruction in local training, other methods might be utilized, such as data augmentation and obfuscation. However, neighboring nodes would equally have access to shared model parameters. Therefore, combining DP techniques with secure aggregation and additive homomorphic encryption is a potential option that acts as a countermeasure in DFL. These strategies can be applied to more federation participants, depending on the application context and the desired level of security.

In other situations, it is preferable to rely on anomaly detection techniques or model misbehavior detection techniques to find harmful players that could harm DFL performance. In contrast to the former, which focuses on the data fed into the model, the latter examines the model's internal performance in light of the task at hand. In certain situations, detection might entail the use of mitigation to lessen the impact of a cyberattack. Last but not least, DFL makes extensive use of cryptographic techniques like secure multiparty computation (SMC) to protect the privacy of communications between nearby nodes. Before sending messages through the federation, members must encrypt them, perform operations on them, and then decrypt the output of those operations.

Despite the security, nodes may experience extremely high computational cost, much like in earlier DP methods. New strategies have therefore been developed to offer appropriate security performance while preserving efficient resource utilization. Deploying a Trusted Execution Environment (TEE), such as Intel SGX processors, is one such strategy that can guarantee the security of the code and data loaded inside of it. This setting enables the secure addition of model parameters during federation while enhancing the trustworthiness of each node in the network. DFL requires procedures that guarantee the decentralization of the architecture while assuring inter-participant reliability in addition to cryptographic tools and safe settings. Also, it can securely store private keys and other sensitive data used in cryptographic operations, such as homomorphic encryption and secure multiparty computation. For communication among nodes, TEE enables secure connections between the peer devices participating in the DFL process, by providing a secure channel for data exchange [30].

5.7.2 Defense methods not applicable to DFL ecosystems

While many of the defense methods used in CFL can be applied to DFL, there are some methods that may not be directly applicable to DFL. Here are a few examples: Secure aggregation: While secure aggregation can be used in DFL, the process of aggregating data across multiple devices or nodes can be more complex than in CFL. This is because DFL typically involves a larger number of nodes, which can make it more difficult to ensure that all nodes are participating honestly and that the aggregated data is accurate. Model training methods: Some model training methods that rely on centralized aggregation of data, such as federated distillation, may not be directly applicable to DFL. This is because the process of aggregating data across multiple nodes in a decentralized environment can be more difficult to secure. Client identification: Client identification methods used in CFL, such as anomaly detection or machine learning models to identify malicious clients,

may be less effective in DFL. This is because DFL typically involves a larger number of clients, which can make it more difficult to identify malicious actors or outliers. Overall, even if many of the same defense strategies utilized in CFL can also be used in DFL, the particular difficulties of a decentralized environment might require some adaptation or the creation of new strategies that are uniquely suited to DFL.

5.7.3 Unique defense methods in DFL ecosystem

There are several unique defense methods in DFL compared to CFL. One of the main differences is that in DFL, each node is responsible for its own security, as opposed to relying on a centralized server in CFL. This means that each node needs to have its own security measures in place to ensure the integrity and confidentiality of its data.

One unique defense method in DFL is the use of Distributed Ledger Technology based defense approach like blockchain technology to ensure the authenticity and integrity of the data being shared between nodes. DLT is a decentralized digital system for keeping track of assets and recording business network interactions. It offers a transparent and unchangeable record of every transaction, handled and validated by a network of users rather than a single authority. A special kind of DLT called blockchain employs encryption to produce secure, unchangeable records of transactions that are grouped into blocks and connected in a chain. A tamper-proof record of all transactions in the chain is made possible by the fact that each block includes a cryptographic hash of the one before it.

Using blockchain-based smart contracts to control communication between nodes is one method DLT can be applied in DFL. A smart contract could be used, for instance, to enforce data access regulations and guarantee that nodes only access data that they are permitted to use. Consensus techniques like Proof of Stake (PoS) or Practical Byzantine Fault Tolerance (PBFT) can also be employed with DLT to make sure that all nodes in the network agree on the state of the shared ledger. Due to its decentralized nature, DFL is more suitable for DLT applications than CFL. The nodes in DFL are already distributed, making it simple for them to host their own blockchain nodes. Additionally, the usage of DLT can add an extra degree of security and accountability because DFL frequently involves sensitive data and needs a high level of trust between nodes. The system's complexity could rise as a result of the additional infrastructure and resources needed to deploy DLT in DFL [32].

By using a decentralized, immutable ledger, blockchain can prevent malicious nodes from tampering with data or introducing false data into the network. So, one of the main benefits of using DLT in DFL is its ability to mitigate data poisoning attacks. Each participant's data contributions can be tracked by the participants, and if any malicious material is found, its source can be determined and the proper actions can be taken. Model poisoning attacks and inference attacks can also be mitigated using DLT. The participants can make sure that the model updates are genuine and haven't been tampered with and can ensure that the sensitive information is kept confidential and is not exposed to unauthorized parties. In addition, in [31] Minfeng introduced that using blockchain can motivate data owners to participate in and stay within an FL ecosystem by continuously contributing their data to the FL model.

Another unique defense method in DFL is the use of reputation systems to identify and exclude malicious nodes from the network. Reputation systems can track the behavior of each node, such as the accuracy of its data contributions and the frequency of its participation, and use this information to determine if a node is trustworthy or not. Nodes with a poor reputation can be excluded from the network or have their contributions weighted less heavily [33]. There are different types of reputation systems with different algorithms that can be used in DFL. One of them is the EigenTrust algorithm, which computes trust scores for each node based on its network of relationships. And the SybilGuard algorithm is another method that employs social network analysis to identify and stop Sybil assaults, in which an attacker establishes numerous false identities to take over the system.

Various DFL attacks, such as Sybil attacks, data poisoning attacks, and model poisoning attacks, can be mitigated with the use of reputation systems. Reputation management solutions can help in preserving the integrity and accuracy of the entire model by spotting and eliminating rogue nodes.

5.8 Open Challenges & Future Directions

Although federated learning has shown great potential in solving privacy issues in machine learning, there are still several challenges that need to be addressed to improve its privacy-preserving capabilities. The primary issues that future DFL solutions may take into consideration are represented by the following points, which are based on the current state of the art. Finding the ideal balance between privacy and performance is one of the primary difficulties in privacy-preserving federated learning. Stronger privacy protections can protect sensitive data, but they may also cause the trained model's accuracy to decline. Therefore, techniques that can guarantee robust privacy while preserving the usefulness of the trained model must be developed. Another challenge is scalability. Even though DFL mitigates this problem in some way by decentralizing the topology, scalability is still a significant challenge. Privacy-preserving techniques can add computational and communication overhead, which can make it difficult to scale the system to handle large datasets.

However, in [34], Boesch pointed out that the most important requirement for privacy-preserving techniques is to provide strict privacy guarantees without over-sacrificing accuracy. Therefore, these techniques must be low-cost in terms of computing, effective in terms of communication, and tolerant of dropped devices. Indeed, state-of-the-art of current privacy-preserving federated learning systems typically consists of traditional cryptographic protocols like SMC and differential privacy. And those techniques impose significant performance overheads and reduce the accuracy of models.

As for future work, FL has a set of security challenges that need further review and research. Although we did introduce a lot of countermeasures for each attack, we only discussed them in theory. In practice, FL can face a lot more challenges than we expected. Firstly, datasets in practice are often more complex and diverse than those in theoretical models, which can lead to performance degradation and loss of accuracy when using

cryptographic techniques. Secondly, in practice, attackers can employ more sophisticated and stealthy attacks, such as model pruning and model spoofing. These attack methods may be difficult to defend against by traditional cryptographic techniques.

Therefore, the defence of FL in practical applications still needs further research and improvement. More sophisticated and advanced defence techniques need to be developed to provide stronger privacy protection and defence against attacks. In addition, further research is needed to translate the methods in theory into practical and workable solutions and to verify their effectiveness in practice.

5.9 Conclusion

Ensuring security and privacy in federated learning (FL) is crucial. However, research in this field is still limited, and the issue of adversarial attacks in distributed federated learning (DFL) is largely unexplored. So, we present a literature review on CFL and DFL, covering recent attacks and defenses. Our present work has answered the following research questions:

Q1. What are the fundamentals of CFL and DFL, and what are the differences? Section III of this report examines the fundamentals of CFL and DFL and highlights their differences. CFL, being reliant on a central server, is susceptible to challenges such as central server bottleneck and single point of attack. In contrast, DFL inherits some advantages of distributed networks, but it also presents new challenges such as the absence of a strong central server to coordinate the federation and the facilitation of attackers through distributed aggregation nodes. Additionally, CFL and DFL face common challenges such as costly communication and significant heterogeneity.

Q2. What are the most recent and representative attacks in CFL? Section IV of the report presents a thorough analysis of the location-based taxonomy of attacks in CFL, encompassing client-side, server-side, and federation-side attacks. This section highlights the most recent and representative types of attacks in CFL and elaborates on where they occur and how they disrupt the model performance or privacy of CFL. Furthermore, the analysis serves as a basis for comparing attacks in DFL.

Q3. How do those attacks affect DFL, and what are the connections and differences? Section V focuses on analyzing the differences and connections between DFL and CFL, with reference to the attacks in CFL. The analysis points out that while DFL has the advantage of avoiding all attacks on the server and being more fault-tolerant, it also exposes new vulnerabilities when faced with certain attacks. In particular, the report provides a detailed examination of the weaknesses exposed by DFL in the event of malware, poisoning, free-riding, sybil, and eclipse attacks.

Q4. What are the popular defense methods in CFL? Section VI provides a comprehensive understanding of the popular defense methods in CFL, and classifies them based on the technical characteristics of the defense methods and their purpose of application, including Cryptographic approaches, Aggregation algorithms, Client Identification and Model

Training. Furthermore, we will also discuss the drawbacks of each method separately to provide a balanced perspective.

Q5. Are these defense methods applicable to DFL, and how are they applied? Section VII illustrates a deeper understanding of the relationship between defense methods used in CFL and DFL. We found that most of the methods can be inherited from CFL but need further modification according to topologies. In addition, there are some special algorithms like Blockchain that can be used to mitigate various attacks.

Bibliography

- [1] Duan, Y., Edwards, J. & Dwivedi, Y. Artificial Intelligence for Decision Making in the Era of Big Data – evolution, Challenges and Research Agenda. *International Journal Of Information Management*. **48** pp. 63-71 (2019,10), <https://www.sciencedirect.com/science/article/pii/S0268401219300581>
- [2] Varghese, B., Wang, N., Barbhuiya, S., Kilpatrick, P. & Nikolopoulos, D. Challenges and Opportunities in Edge Computing. *2016 IEEE International Conference On Smart Cloud (SmartCloud)*. (2016,11), <https://ieeexplore.ieee.org/abstract/document/7796149/>
- [3] Shi, W., Cao, J., Zhang, Q., Li, Y. & Xu, L. Edge Computing: Vision and Challenges. *IEEE Internet Of Things Journal*. **3** pp. 637-646 (2016,10)
- [4] Mammen, P. Federated Learning: Opportunities and Challenges. *ArXiv:2101.05428 [cs]*. (2021,1), <https://arxiv.org/abs/2101.05428>
- [5] Gosselin, R., Vieu, L., Loukil, F. & Benoit, A. Privacy and Security in Federated Learning: A Survey. *Applied Sciences*. **12** pp. 9901 (2022,10)
- [6] Sikandar, H., Waheed, H., Tahir, S., Malik, S. & Rafique, W. A Detailed Survey on Federated Learning Attacks and Defenses. *Electronics*. **12** pp. 260 (2023,1)
- [7] Liu, P., Xu, X. & Wang, W. Threats, attacks and defenses to federated learning: issues, taxonomy and perspectives. *Cybersecurity*. **5** (2022,2)
- [8] Gholami, A., Torkzaban, N. & Baras, J. Trusted Decentralized Federated Learning. *2022 IEEE 19th Annual Consumer Communications & Networking Conference (CCNC)*. (2022,1)
- [9] Bouacida, N. & Mohapatra, P. Vulnerabilities in Federated Learning. (IEEE Access,2021), <https://www.semanticscholar.org/paper/Vulnerabilities-in-Federated-Learning-Bouacida-Mohapatra/4eb8c7023c57f76d5e2401c033e7e2769be03a79>
- [10] Zhang, Z., Cao, X., Jia, J. & Gong, N. FLDetector: Defending Federated Learning Against Model Poisoning Attacks via Detecting Malicious Clients. *Proceedings Of The 28th ACM SIGKDD Conference On Knowledge Discovery And Data Mining*. (2022,8), <https://arxiv.org/pdf/2207.09209.pdf>

- [11] Xie, C., Huang, K., Chen, P. & Li, B. DBA: Distributed Backdoor Attacks against Federated Learning. (openreview.net,2022,2), <https://openreview.net/forum?id=rkgyS0VFvr>
- [12] Bhagoji, A., Chakraborty, S., Mittal, P. & Calo, S. Analyzing Federated Learning through an Adversarial Lens. (PMLR,2019,5), <https://proceedings.mlr.press/v97/bhagoji19a.html>
- [13] Dwork, C. & Roth, A. The Algorithmic Foundations of Differential Privacy. *Foundations And Trends® In Theoretical Computer Science*. **9** pp. 211-407 (2013)
- [14] Wei, K., Li, J., Ding, M., Ma, C., Yang, H., Farhad, F., Jin, S., Quek, T. & Poor, H. Federated Learning with Differential Privacy: Algorithms and Performance Analysis. *ArXiv:1911.00222 [cs]*. (2019,11), <https://arxiv.org/abs/1911.00222>
- [15] Yu, D., Zhang, H., Chen, W., Liu, T. & Yin, J. Gradient Perturbation is Under-rated for Differentially Private Convex Optimization. *ArXiv:1911.11363 [cs, Stat]*. (2020,10), <https://arxiv.org/abs/1911.11363>
- [16] Aronovich, L. IBM® Federated Learning with Homomorphic Encryption. (IBM Data Science in Practice,2023,1), <https://medium.com/ibm-data-ai/ibm-federated-learning-with-homomorphic-encryption-a4cad23f012c>
- [17] Shokri, R., Stronati, M., Song, C. & Shmatikov, V. Membership Inference Attacks against Machine Learning Models. *ArXiv:1610.05820 [cs, Stat]*. (2017,3), <https://arxiv.org/abs/1610.05820>
- [18] Huang, Y., Gupta, S., Song, Z., Li, K. & Arora, S. Evaluating Gradient Inversion Attacks and Defenses in Federated Learning. *ArXiv:2112.00059 [cs]*. (2021,11), <https://arxiv.org/abs/2112.00059>
- [19] McSherry, F. & Talwar, K. Mechanism Design via Differential Privacy. *48th Annual IEEE Symposium On Foundations Of Computer Science (FOCS'07)*. (2007), <https://www.semanticscholar.org/paper/Mechanism-Design-via-Differential-Privacy-McSherry-Talwar/f9a751c75d4fe2d3f747166be2cbee78e3910b50>
- [20] Mugunthan, V. & Polychroniadou, A. SMPAI: Secure Multi-Party Computation for Federated Learning. (www.semanticscholar.org,2019)
- [21] Bonawitz, K., Ivanov, V., Kreuter, B., Marcedone, A., McMahan, H., Patel, S., Ramage, D., Segal, A. & Seth, K. Practical Secure Aggregation for Federated Learning on User-Held Data. (Google Research,2016), <https://research.google/pubs/pub45808/>
- [22] Hong, C. Federated learning and MPC. (The Sugar Beet: Applied MPC,2020,1), <https://medium.com/applied-mpc/federated-learning-and-mpc-93ab84c7909a>
- [23] Zhu, L., Liu, Z. & Han, S. Deep Leakage from Gradients. *ArXiv:1906.08935 [cs, Stat]*. (2019,12), <https://arxiv.org/abs/1906.08935>

- [24] Pillutla, K., Kakade, S. & Harchaoui, Z. Robust Aggregation for Federated Learning. *ArXiv:1912.13445 [cs, Stat]*. (2019,12), <https://arxiv.org/abs/1912.13445>
- [25] Rey, V., Sánchez Sánchez, P., Huertas Celdrán, A. & Bovet, G. Federated learning for malware detection in IoT devices. *Computer Networks*. **204** pp. 108693 (2022,2)
- [26] Seo, H., Park, J., Oh, S., Bennis, M. & Kim, S. Federated Knowledge Distillation. *ArXiv:2011.02367 [cs, Math]*. (2020,11), <https://arxiv.org/abs/2011.02367>
- [27] Zhang, J., Wu, D., Liu, C. & Chen, B. Defending Poisoning Attacks in Federated Learning via Adversarial Training Method. *Communications In Computer And Information Science*. pp. 83-94 (2020,11)
- [28] Mothukuri, V., M. Parizi, R. & Pouriyeh, S. A survey on security and privacy of federated learning. *Future Generation Computer Systems*. **115** pp. 619-640 (2021,2), <https://www.sciencedirect.com/science/article/abs/pii/S0167739X20329848>
- [29] Beltrán, E., Pérez, M., Sánchez, P., Bernal, S., Bovet, G., Pérez, M., Pérez, G. & Celdrán, A. Decentralized Federated Learning: Fundamentals, State-of-the-art, Frameworks, Trends, and Challenges. *ArXiv:2211.08413 [cs]*. (2023,4), <http://arxiv.org/abs/2211.08413>
- [30] Mo, F., Haddadi, H., Katevas, K., Marin, E., Perino, D. & Kourtellis, N. PPFL: privacy-preserving federated learning with trusted execution environments. *Proceedings Of The 19th Annual International Conference On Mobile Systems, Applications, And Services*. (2021,6)
- [31] Qi, M., Wang, Z., Chen, S. & Xiang, Y. A Hybrid Incentive Mechanism for Decentralized Federated Learning. *Distributed Ledger Technologies: Research And Practice*. **1** pp. 1-15 (2022,9)
- [32] Li, J., Shao, Y., Wei, K., Ding, M., Ma, C., Shi, L., Han, Z. & Poor, H. Blockchain Assisted Decentralized Federated Learning (BLADE-FL): Performance Analysis and Resource Allocation. *ArXiv:2101.06905 [cs]*. (2021,5), <https://arxiv.org/abs/2101.06905>
- [33] Chen, X., Wang, T. & Zhang, S. The Design of Reputation System for Blockchain-based Federated Learning. (IEEE Xplore,2021,12), <https://ieeexplore.ieee.org/document/9694014>
- [34] Boesch, G. An Introduction to Federated Learning: Challenges and Applications. (viso.ai,2022,7), <https://viso.ai/deep-learning/federated-learning/>
- [35] Heilman, E., Kendler, A., Zohar, A. & Goldberg, S. Eclipse Attacks on Bitcoin's Peer-to-Peer Network. (2015), <https://www.usenix.org/system/files/conference/usenixsecurity15/sec15-paper-heilman.pdf>
- [36] Zhang, C., Xie, Y., Bai, H., Yu, B., Li, W. & Gao, Y. A survey on federated learning. *Knowledge-Based Systems*. **216** pp. 106775 (2021,3)

- [37] Pang, J., Huang, Y., Xie, Z., Han, Q. & Cai, Z. Realizing the Heterogeneity: A Self-Organized Federated Learning Framework for IoT. *IEEE Internet Of Things Journal*. **8** pp. 3088-3098 (2021,3), <https://ieeexplore.ieee.org/document/9134408>
- [38] Li, T., Sahu, A., Talwalkar, A. & Smith, V. Federated Learning: Challenges, Methods, and Future Directions. *IEEE Signal Processing Magazine*. **37** pp. 50-60 (2020,5)
- [39] Ye, H., Liang, L. & Li, G. Decentralized Federated Learning With Unreliable Communications. *IEEE Journal Of Selected Topics In Signal Processing*. **16** pp. 487-500 (2022,4)
- [40] Abdulrahman, S., Tout, H., Ould-Slimane, H., Mourad, A., Talhi, C. & Guizani, M. A Survey on Federated Learning: The Journey From Centralized to Distributed On-Site Learning and Beyond. *IEEE Internet Of Things Journal*. **8** pp. 5476-5497 (2021,4)
- [41] Abdellatif, A., Mhaisen, N., Mohamed, A., Erbad, A., Guizani, M., Dawy, Z. & Nasreddine, W. Communication-efficient hierarchical federated learning for IoT heterogeneous systems with imbalanced data. *Future Generation Computer Systems*. **128** pp. 406-419 (2022,3)
- [42] Brendan, M., Moore, E., Ramage, D., Hampson, S. & Arcas, B. Communication-Efficient Learning of Deep Networks from Decentralized Data. (arXiv.org,2016), <https://arxiv.org/abs/1602.05629>
- [43] Basu, D., Data, D., Karakus, C. & Diggavi, S. Qsparse-local-SGD: Distributed SGD with Quantization, Sparsification and Local Computations. (Curran Associates, Inc.,2019), <https://proceedings.neurips.cc/paper/2019/hash/d202ed5bcfa858c15a9f383c3e386ab2-Abstract.html>
- [44] Li, T., Sahu, A., Zaheer, M., Sanjabi, M., Talwalkar, A. & Smith, V. Federated Optimization in Heterogeneous Networks. *ArXiv:1812.06127 [cs, Stat]*. (2020,4), <https://arxiv.org/abs/1812.06127>
- [45] Wang, H., Kaplan, Z., Niu, D. & Li, B. Optimizing Federated Learning on Non-IID Data with Reinforcement Learning. (IEEE Xplore,2020,7), <https://ieeexplore.ieee.org/abstract/document/9155494>
- [46] Goetz, J., Malik, K., Bui, D., Moon, S., Liu, H. & Kumar, A. Active Federated Learning. *ArXiv:1909.12641 [cs, Stat]*. (2019,9), <https://arxiv.org/abs/1909.12641>
- [47] Bonawitz, K., Eichner, H., Grieskamp, W., Huba, D., Ingerman, A., Ivanov, V., Kiddon, C., Konečný, J., Mazzocchi, S., Brendan, M., Overveldt, V., Petrou, D., Ramage, D. & Roselander, J. Towards Federated Learning at Scale: System Design. (arXiv.org,2019), <https://arxiv.org/abs/1902.01046>
- [48] Zeng, R., Zeng, C., Wang, X., Li, B. & Chu, X. A Comprehensive Survey of Incentive Mechanism for Federated Learning. *ArXiv:2106.15406 [cs]*. (2021,6), <https://arxiv.org/abs/2106.15406>

- [49] Konečný, J., McMahan, H., Yu, F., Richtárik, P., Suresh, A. & Bacon, D. Federated Learning: Strategies for Improving Communication Efficiency. *ArXiv:1610.05492 [cs]*. (2017,10), <https://arxiv.org/abs/1610.05492>
- [50] Jeong, E., Oh, S., Kim, H., Park, J., Bennis, M. & Kim, S. Communication-Efficient On-Device Machine Learning: Federated Distillation and Augmentation under Non-IID Private Data. *ArXiv:1811.11479 [cs, Stat]*. (2018,11), <https://arxiv.org/abs/1811.11479>
- [51] Yu, T., Bagdasaryan, E. & Shmatikov, V. Salvaging Federated Learning by Local Adaptation. *ArXiv:2002.04758 [cs, Stat]*. (2020,2), <https://arxiv.org/abs/2002.04758>
- [52] Wang, S., Lee, M., Hosseinalipour, S., Morabito, R., Chiang, M. & Brinton, C. Device Sampling for Heterogeneous Federated Learning: Theory, Algorithms, and Implementation. (IEEE Xplore,2021,5), <https://ieeexplore.ieee.org/abstract/document/9488906>
- [53] Zheng, C., Ali, A., Zawad, S., Truex, S., Anwar, A., Baracaldo, N., Zhou, Y., Ludwig, H., Yan, F. & Cheng, Y. TiFL: A Tier-based Federated Learning System. *ArXiv (Cornell University)*. (2020,6)
- [54] He, L., Bian, A. & Jaggi, M. COLA: Decentralized Linear Learning. *ArXiv:1808.04883 [cs, Stat]*. (2019,6), <https://arxiv.org/abs/1808.04883>
- [55] Wang, T., Liu, Y., Zheng, X., Dai, H., Jia, W. & Xie, M. Edge-Based Communication Optimization for Distributed Federated Learning. *IEEE Transactions On Network Science And Engineering*. **9** pp. 2015-2024 (2022,7), <https://ieeexplore.ieee.org/document/9446648>
- [56] Kang, J., Xiong, Z., Niyato, D., Zou, Y., Zhang, Y. & Guizani, M. Reliable Federated Learning for Mobile Networks. *ArXiv:1910.06837 [cs]*. (2019,10), <https://arxiv.org/abs/1910.06837>
- [57] Bedi, A., Koppel, A. & Rajawat, K. Asynchronous Online Learning in Multi-Agent Systems With Proximity Constraints. *IEEE Transactions On Signal And Information Processing Over Networks*. **5** pp. 479-494 (2019,9), <https://ieeexplore.ieee.org/abstract/document/8656546>
- [58] Pappas, C., Chatzopoulos, D., Lalis, S. & Vavalis, M. IPLS: A Framework for Decentralized Federated Learning. (IEEE Xplore,2021,6), <https://ieeexplore.ieee.org/abstract/document/9472790>
- [59] Jere, M., Farnan, T. & Koushanfar, F. A Taxonomy of Attacks on Federated Learning. *IEEE Security & Privacy*. **19** pp. 20-28 (2021,3)
- [60] Zhang, J., Zhang, J., Chen, J. & Yu, S. GAN Enhanced Membership Inference: A Passive Local Attack in Federated Learning. (IEEE Xplore,2020,6), <https://ieeexplore.ieee.org/abstract/document/9148790>

- [61] Lyu, L., Yu, H. & Yang, Q. Threats to Federated Learning: A Survey. *ArXiv:2003.02133 [cs, Stat]*. (2020,3), <https://arxiv.org/abs/2003.02133>
- [62] Tolpegin, V., Truex, S., Gursoy, M. & Liu, L. Data Poisoning Attacks Against Federated Learning Systems. *Computer Security – ESORICS 2020*. pp. 480-501 (2020)
- [63] Bagdasaryan, E., Veit, A., Hua, Y., Estrin, D. & Shmatikov, V. How To Backdoor Federated Learning. *ArXiv:1807.00459 [cs]*. (2019,8), <https://arxiv.org/abs/1807.00459>
- [64] Fraboni, Y., Vidal, R. & Lorenzi, M. Free-rider Attacks on Model Aggregation in Federated Learning. *ArXiv:2006.11901 [cs, Stat]*. (2021,2), <https://arxiv.org/abs/2006.11901>
- [65] Lin, J., Du, M. & Liu, J. Free-riders in Federated Learning: Attacks and Defenses. *ArXiv:1911.12560 [cs, Stat]*. (2019,11), <https://arxiv.org/abs/1911.12560>
- [66] Biggio, B., Corona, I., Maiorca, D., Nelson, B., Šrndić, N., Laskov, P., Giacinto, G. & Roli, F. Evasion Attacks against Machine Learning at Test Time. *Advanced Information Systems Engineering*. pp. 387-402 (2013)
- [67] Ayub, M., Johnson, W., Talbert, D. & Siraj, A. Model Evasion Attack on Intrusion Detection Systems using Adversarial Machine Learning. (IEEE Xplore,2020,3), <https://ieeexplore.ieee.org/abstract/document/9086268>
- [68] Li, J., Lyu, L., Liu, X., Zhang, X. & Lyu, X. FLEAM: A Federated Learning Empowered Architecture to Mitigate DDoS in Industrial IoT. *IEEE Transactions On Industrial Informatics*. **18** pp. 4059-4068 (2022,6), <https://ieeexplore.ieee.org/abstract/document/9454328>
- [69] Bhushan, B., Sahoo, G. & Rai, A. Man-in-the-middle attack in wireless and computer networking — A review. *2017 3rd International Conference On Advances In Computing, Communication & Automation (ICACCA) (Fall)*. (2017,9), <https://ieeexplore.ieee.org/document/8344724>
- [70] Derui, Wang, Li, C., Wen, S., Nepal, S. & Xiang, Y. Man-in-the-Middle Attacks against Machine Learning Classifiers via Malicious Generative Models. *ArXiv:1910.06838 [cs]*. (2019,10), <https://arxiv.org/abs/1910.06838>
- [71] Fredrikson, M., Jha, S. & Ristenpart, T. Model Inversion Attacks that Exploit Confidence Information and Basic Countermeasures. *Proceedings Of The 22nd ACM SIGSAC Conference On Computer And Communications Security - CCS '15*. (2015), <https://rist.tech.cornell.edu/papers/mi-ccs.pdf>
- [72] Hitaj, B., Ateniese, G. & Perez-Cruz, F. Deep Models Under the GAN: Information Leakage from Collaborative Deep Learning. *ArXiv:1702.07464 [cs, Stat]*. (2017,9), <https://arxiv.org/abs/1702.07464>
- [73] Gao, J., Wang, W., Liu, Z., Billah, M. & Campbell, B. Decentralized Federated Learning Framework for the Neighborhood. *International Conference On Embedded Networked Sensor Systems*. (2021,11)

- [74] Zhao, S., Wu, Y., Sun, R., Qian, X., Zi, D., Xie, Z., Tong, E., Niu, W., Liu, J. & Han, Z. Blockchain-based decentralized federated learning: A secure and privacy-preserving system. *2021 IEEE 23rd Int Conf On High Performance Computing & Communications; 7th Int Conf On Data Science & Systems; 19th Int Conf On Smart City; 7th Int Conf On Dependability In Sensor, Cloud & Big Data Systems & Application (HPCC/DSS/SmartCity/DependSys)*. (2021,12)
- [75] Chen, Y., Ning, Y., Slawski, M. & Rangwala, H. Asynchronous Online Federated Learning for Edge Devices with Non-IID Data. *ArXiv:1911.02134 [cs]*. (2020,10), <https://arxiv.org/abs/1911.02134>
- [76] Zhang, K., Liang, X., Lu, R. & Shen, X. Sybil Attacks and Their Defenses in the Internet of Things. *IEEE Internet Of Things Journal*. **1** pp. 372-383 (2014,10)

Chapter 6

The Impact of Open Source on Cybersecurity

Lucas Krauter

Niklas A. Schmatloch

Since open source as a concept emerged and made its way into software development, its importance and popularity has been continuously growing in both private as well as commercial software. Due to its inherent properties, like transparency and reusability it induced fundamental changes in development, collaboration and usage of software in many projects and companies, many of which impact software security. Next to the more apparent direct influence on the quality in terms of security of the software, the impact open source has on the development process, distribution and reuse of software in other projects opens new attack vectors but also possibilities for improvement.

This research work first clarifies the terms open source and cybersecurity, then discusses theoretical and conceptual advantages and risks of open source, investigates several case studies of popular security incidents and finally demonstrates strategies and tools to improve found problems and the security of open source in general.

Contents

6.1	Introduction	143
6.1.1	Introduction Open Source Software	143
6.1.2	Introduction to Cybersecurity	145
6.2	Security Characteristics of Open and Closed Source Software	147
6.2.1	Introduction of Stakeholders	148
6.2.2	Advantages of OSS	148
6.2.3	Risks of OSS	150
6.3	Case Studies of Software incidents	151
6.3.1	Example for Open Source	151
6.3.2	Example for Closed source	154
6.3.3	Lessons learned	154
6.4	Improve security of OSS	155
6.4.1	Create secure software	155
6.4.2	Example Tools	156
6.5	Summary	159

6.1 Introduction

Since many years the popularity and importance of open source software has been increasing - a trend which is not expected to be reverted in the near future. While open source offers advantages that are very apparent, like transparency and the possibility of collaborating easily, exactly this shift towards more collaborative development opens software development also to more hidden dangers and ways security can be harmed.

In order to understand better, that open source is not just a different type of software, but affects software development in a much broader way, this work will first clarify the terms open source and cybersecurity. It will then discuss theoretical and conceptual advantages and risks that this shift brings with it and subsequently investigate impactful real-world security incidents to get a more practical grasp and understanding on common causes, and effectively what one needs to look out for when developing open source software, respectively how one can avoid making the same mistakes.

Finally, this work will present strategies and tools to on the one hand make open source projects more secure, and on the other hand demonstrate how open source software affects the availability of tools like these.

6.1.1 Introduction Open Source Software

Colloquially, the term "open source" can be misunderstood as simply meaning software of which the source code is made public. The usage and definitions of the major stakeholders in open source software, such as companies or the Open Source Initiative itself, however, define the term as requiring additional properties, mostly concerning the licensing and allowed use of the respective software. It therefore is useful to recapitulate briefly the history of origin of open source, the term, and similar paradigms. Figure 2.1 gives a broad overview on how software can be distinguished by openness and commercial use.

Due to intellectual property laws in large parts of the worlds, specifically Europe and the US, any creative work, including software, is protected by copyright. In the early days of software development, commercial software usually was developed by central authorities, such as companies, and distributed with restrictive licenses, preventing unauthorised usage, modification or redistribution. An example for early proprietary software is the operating system UNIX (1969) developed by Bell Laboratories [46].

The first major move away from proprietary software started with the Free Software Movement in the 1970s, materialising in the GNU project (1983) and the Free Software Foundation (FSF) (1985), which formalised the idea of "Free Software". According to the definition of the FSF, Free Software needs to permit four freedoms to its users [20]:

- Run the program as you wish, for any purpose
- Study and change the program (requires source code access)
- Redistribute exact copies
- Distribute modified versions (forks)

Due to the ambiguity of the meaning of the word "Free" it often is also called Libre Software, emphasizing that its pursuit is to ensure the freedom of users without demanding software to be non-commercial [20].

In many regards this can be seen also as the beginning of open source software as Free Software de facto always is open source software as defined later by the open source Initiative (OSI) founded in 1998. In general, the open source movement, even though chronologically succeeding the Free Software movement, is less of a successor of the FSF but rather a split, or in other words, "doing the same thing for different reasons". While the Free Software movements focus is on digital ethics and user freedom in computing, the open source movement aims at utilising the practical advantages of collaborative working with open source software. Similarly to the FSF, the OSI requires software to follow certain criteria to be considered open source [36]:

- **Free Redistribution**

It must be allowed to sell or freely distribute the software.

- **Source Code**

The users of the program must be able to access and read the source code.

- **Derived Works**

Open source software allows a program to be modified by any user and the modified version, just as exact copies (criterion 1), can be sold or given away freely.

- **Integrity of The Author's Source Code**

Alternatively to the previous direct redistribution of modified versions, open source software may also allow the distribution of the exact version together with the possibility of patch files.

- **No Discrimination Against Persons or Groups**

All persons and groups must have the same rights to make use of the program for it to be considered open source.

- **No Discrimination Against Fields of Endeavor**

The license of open source software cannot restrict in which fields the software may be used in. This may include use cases the creator of the software may deem as immoral.

- **Distribution of License**

The users cannot be required to execute additional agreements. This prevents for example non-disclosure agreements.

- **License Must Not Be Specific to a Product**

This prevents that the open source nature of the software is restricted by requiring that it may only be used as part of a larger project.

- **License Must Not Restrict Other Software**

Open source software can be used in combination with any other software and may not restrict this.

- **License Must Be Technology-Neutral**

The license must not require anything that would restrict the ways it can be distributed or executed.

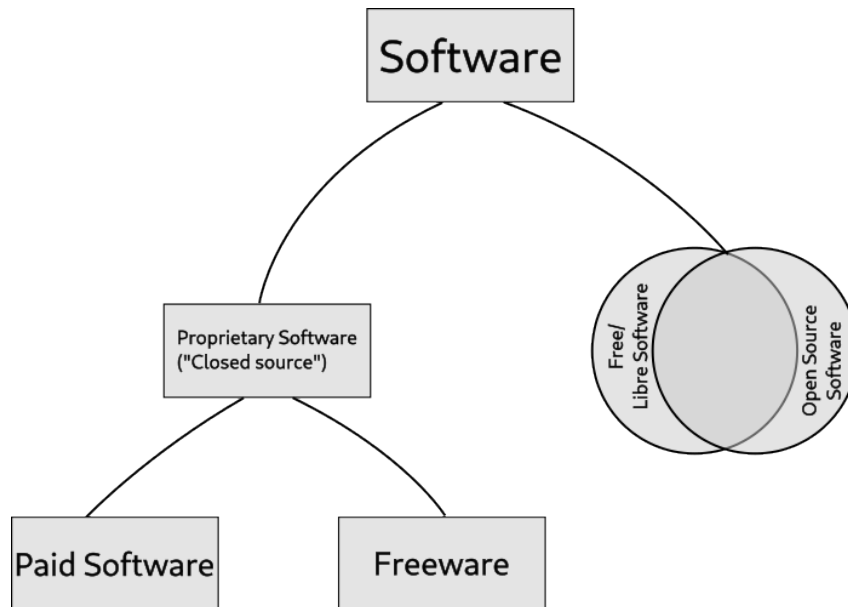


Figure 6.1: Classification of software

From this definition of open source software it becomes clear that the impact the growing popularity open source has in software development affects way more aspects than just the colloquial understanding that the software's source code can be read. While it certainly does include this effect of increased transparency, due to criterion 2, it also changes not just the software itself but also the development process around it. While closed proprietary software usually is developed and maintained by a single stakeholder, for example a company, or closely contractually connected instances, due to the possibility to reuse software (criterion 1) and distribute modifications (criterion 3), this facilitates collaboration between loosely connected developers and stakeholders. The latter also leads software to more easily being built upon and to a lot of projects having a common code base (either due to having a common ancestor or by including the same packages). Furthermore, it also allows abandoned projects to be revived by other developers.

Lastly, aforementioned increase in collaboration between developers, many of which are private persons, and the decrease of contractual regulation between these developers often also leads to less legal accountability than in conventional software development.

6.1.2 Introduction to Cybersecurity

The introduction to cybersecurity should start with a definition of the term. It is shared across many different domains, so that it is impossible to state a single definition. However, this paragraph will try to summarize the important points. In an abstract way, cybersecurity is often viewed as a subdomain of IT Security and that itself can be seen as a subtopic of the Information Technology Sector. A more theoretical definition could

sound as the following: cybersecurity refers to a state of a Computer System in which it is protected against attacks of any kind from the cyber domain [47]. A Computer System means some devices that process data and are interconnected. An attack can be described in abstract terms as an event with bad intentions from the internet, which interacts with the target system. It can have multiple forms, so that it can reach from of an unauthorized access to electronic data to a large scale surveillance of internet traffic of civilians. As the range of those attacks is literally limitless and only restricted by one's fantasies, this work will mention a few examples to a later point on a high level. The term cybersecurity also includes the measurements to defend a system from the internet, which also takes offensive methods into account [48]. Examples for those will also be mentioned later.

The next part of this chapter will have a closer look at different kinds of cyber criminals. These are actors on the internet which do not shy away from committing crimes in order to promote their interest on the internet. First, it is possible to classify the actors and their motivations into different groups based on their professional level as seen in the picture below:

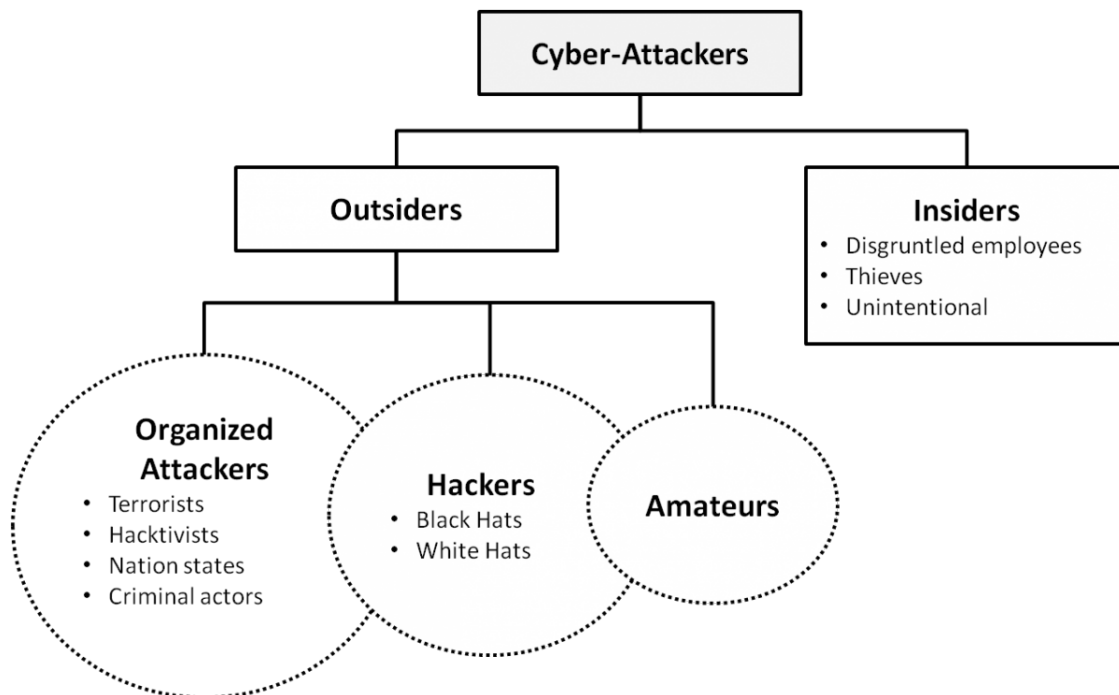


Figure 6.2: Categories of cyber-attackers [15]

This chapter will focus on the outsiders and ignore the insiders, as this separation doesn't make sense with open source created systems.

Organized attackers: The organized attackers can be defined as a group of people with a shared target system. In most cases those attackers can launch highly sophisticated attacks against their victims as they have high resources. Actors such as terrorists or nation states promote their own interests with their attacks. Mostly these are motivations beyond monetarily such as the defence of their own citizens or standing up for their beliefs such as religious ones. Sometimes the classification of these two depends on the own standpoint [15].

The Sony Picture hack in 2014 serves as a good example. There Sony was attacked by the hacker group Guardians of Peace, who blackmailed Sony into not releasing the movie "The Interview". The movie was mocking North Korea and its leader Kim Jong-un. After some research, the FBI was able to find proofs that the attack originated from North Korea as a retaliation [2].

Hactivists are protagonists who participate in cyber activism. They form groups and attack targets in the name of social or political causes. One of the most famous examples is the group Anonymous, who also participates in a cyberwar against Russia by causing damage to target systems within Russia [23].

The fourth group of **criminal actors** often have a monetary motivation in mind. These are criminal organizations who focus on cyber crimes to enrich themselves. A famous example is the Ransomware groups Lockbit. This group of people operates a so-called ransomware-as-a-service. They provide a ransomware-malware which can be purchased by other criminals to deploy against targets. Through different threats the target gets blackmailed into paying some ransom to the criminals. The software provider will receive a share of the ransom. This makes it a profitable business for both parties involved. It is important to note that obviously LockBit is actively encouraging to use the software for a malicious purpose which in general criminalises the provision of ransomware-as-a-service [7].

Hackers: These type of actors can be experts but lack of resources. In most cases they do not launch large sized attacks but focus on outsmarting the target system. Depending on their attack motivation these people can be categorized into the two groups Black Hats and White Hats. Black Hats share the same motivation as criminal actors, they mostly want to extract money from the attack. The goal of White Hats is to scan the target system for weak points. The result will be communicated with the target system's owner in exchange for some monetary compensation. This task can be either done via a direct connection, e.g. hire a team of security experts, or via an open announcement. Therefore, the White Hats are more or less hired by the company to check the defence of the own system. The important distinction to Black Hats is that White Hats always move within the legal restrictions and do not intentionally commit any harm to the target systems. An example for working titles in the field of White Hats are Security experts [15].

Amateurs: These class of cyber criminals do not have the expertise of Hackers. However, tools are often available on the internet as well as directives how to use them. Therefore, it is possible for a wider group of people to launch attacks against target systems, which can cause significant damage. The motivation of these amateurs can have a wide range from curiosity to monetary advantages [15].

6.2 Security Characteristics of Open and Closed Source Software

In the previous chapter, different kinds of attackers were discussed. This chapter introduces the different characteristics of users of open source software. For a better overview,

the different types of users are first defined and explained. Then the advantages and risks will be explained from their point of view.

6.2.1 Introduction of Stakeholders

First, the different viewpoints on OSS will be introduced. Following the production chain of OSS, the **developer** is the first. If this article refers to a developer of OSS at a later point, it always means the creator of the source code. A developer seeks to offer their software in the best version possible. Therefore, it is well maintained and the most secure according to their abilities. Most creators also reach out to the community and search for volunteers to help maintain the software.

The next step is the **user** of the open source software. In this case, the user wants to adopt the functionality of the OSS for their own advantages. This can be in multiple forms, e.g.: using an OSS Tool for increasing their productivity or building their own software on top of an OSS library. The user generally wants to adopt the advantages of the OSS, but this always includes disadvantages or risks of the given software. Hence, users always try to maximize the positive and minimize the negative aspects.

The third angle looks at the software from an **attacker** point of view. Users with this viewpoint can be separated into the before explained classes of white and black hat hackers. The goal is trying to find weak spots in the open source software. Found vulnerabilities in the software are then used depending on the intention of the explorer.

It is important to note that people from all of these three categories can each have good or bad intentions. For example, a user of an OSS could be an amateur cyber criminal downloading a tool and trying to cause some damage to a target.

6.2.2 Advantages of OSS

In this chapter, the advantages of using open source software in cybersecurity is discussed. The advantages are sorted after the different point of views from the stakeholders.

First, the following paragraph is about the developer point of view. The idea is to see what security benefits can be gained by licensing and releasing the self developed software as open source. If the source code is shared in the open, it is available for everyone to evaluate. Most times, the process of addressing security issues is also made public when open up the software to the community. This creates transparency of how well this patch process works. These public artefacts will build more trust for the users, because they can assess the security of the software they are using.

The inspector can report their findings of security issues or even fix it themselves and report it to the maintainer. Thus, the security of the software will be assessed and fixed by multiple users of different backgrounds. This allows the software to be protected against a broader range of attack vectors than a single developer could have anticipated. Besides, a community of maintainers can remove security issues significantly faster and react faster to new threats [18].

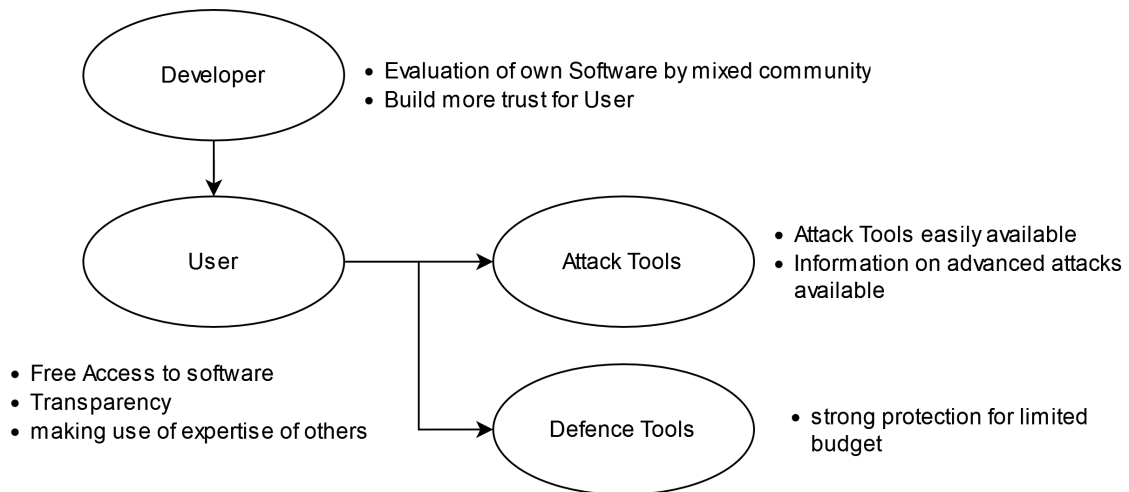


Figure 6.3: Overview and Classification of Benefits

Also, another process made public is the managing of dependencies of the project. Statistically, most Commercial software already uses open source to a great factor. If one of the used libraries happens to have a security vulnerability and needs to be updated this can be addressed faster by a community, where everyone can check and report whether the software has to update its dependencies. This creates again more transparency and more secure software. In commercial software such updates are likely to postponed as long as possible. No one has the chance to call out the creators, until someone tries to exploit this issue [38] .

The most important consequence of the source code made public is the higher trust of the user into the software. This leads to a clear competitive advantage over commercial software offers, which results into more users picking this software.

Second, the most direct advantages can be seen from the user view. Through using a well maintained open source software, the user has access to software which might be licensed otherwise. Next to the source code the creation process of the software is also often made public, so that there is a lot of transparency of how the software was written and how well it is maintained. As a consequence the user can easily choose the most secure option from different software solutions in a specific context. By using OSS, the user can make use of the expertise of the developers. Moreover, these tools are mostly free to use, thus lowering the usage costs significantly. Another advantage is the open collaboration on these tools. If the user needs a new feature, they can implement it themselves and share it back with the creator [27].

The attacker site is about deploying attacks with open source software. Complete computer operating systems (OS) with preinstalled multiple attack tools are available as open source software. The most famous example is the distribution Kali Linux [50]. There is an image for the most common computer architectures and a well maintained documentation for installation instructions which makes it straightforward to install and use such tools. For more advanced users which might look into attacks which are not widely known, a lot of documentation can be found online. An example is vx-underground which claim that they have "the largest collection of malware source code, samples, and papers on the internet" [52]. These are public available making it easy to download by everyone.

On the defence site, this allows even small enterprises to implement a strong protection against cyberattacks without spending too much on building expertise or software licences. This is especially the case, for open source libraries for security mechanisms such as encryption. If one wants to safely implement an encryption protocol, it is highly likely that there are several vulnerabilities. In those cases, when the expertise is missing it is better to fall back on open source implementations created by others [27].

6.2.3 Risks of OSS

First, this chapter will have a closer look at the security risks that can occur within an open source project.

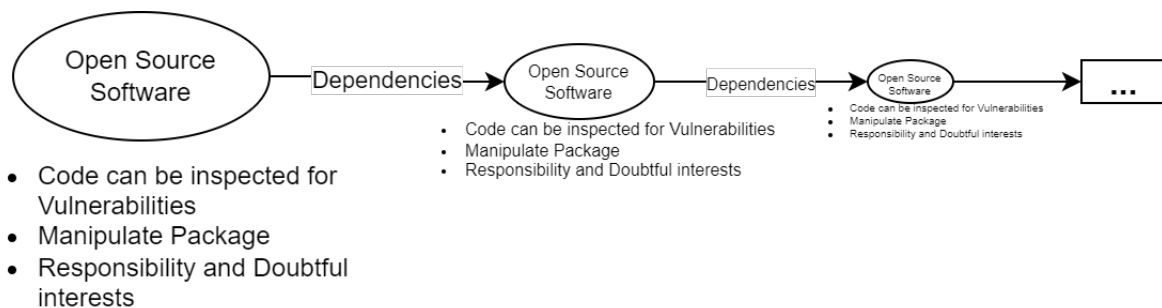


Figure 6.4: Overview of Open Source Software Risks

The most obvious is that everyone can have a look at the software and scan it for security issues. If an inspector found a vulnerability they might not report it directly to developers responsible for fixing the issue. Instead, they might try to exploit the vulnerability or sell it to any attackers with bad intentions.

Next to scanning a package for a vulnerability, there might also be the attack vector to simply introduce an issue into the code on purpose. This can be done either by the responsible developer or by bad actors taking over the project. In both cases, this will have a security impact for the user of this OSS [35].

Another one of the key questions is the responsibility of the projects. Most OSS is licensed so that solely the user is responsible for any damages on the system introduced by using the public code. Therefore, the developer can not be held liable for any security vulnerabilities in his published code. This leads to many unmaintained and outdated open source projects as no one claims responsibility for it [35].

Another important point is that these employed open source projects have dependencies which are likely part of the open source world themselves. Thus, the system's defence is only as strong as the weakest used software in the link. Very often those chains are too long to still have the overview over every dependency of a project.

To sum up, for the user deploying OSS might come with a big trade-off. The open source

code or its dependencies might have vulnerabilities. Through using it the user might inherit those and make their system vulnerable against attacks from the outside. It is no wonder that this issues is also on the OWASP Top 10 list from 2021 [32].

From a perspective of the developer the enforced transparency in OSS also comes with costs of possible attacks on the developer's privacy. Since the complete development process is openly available the metadata can be used to de-anonymise the profile. However, since this a security threat to a single person rather than an enterprise, this work will not go into further detail about these issues [19].

In contrast to the chapter on the benefits of OSS attack tools, this paragraph will highlight the risks coming along with it. Since the knowledge and the software is public available on the internet, anyone can access and leverage those. It is best explained and showed on the basis of the example of Anonymous. They published several tools to deliver Denial of Service attacks. In those attacks, the target gets overwhelmed with network packages. Those tools are available online and distributed to followers of anonymous. In a call for action these followers attack targets at the same time to take down websites of their declared enemies. The risk here is that most users are not aware that those tools do not operate anonymously without further techniques to hide the origin of the user. Therefore, attacks are traceable back to the user, which is legally responsible for his illegal doings on the internet [1].

Moreover, just as a side note that there is not even remotely an age verification before gaining access to such tools. Hence, even youth people have access and are not protected against wrong doings.

To sum up, many users employ such OSS tools without fully understanding the consequences of their actions. Thus, the questions arises whether these tools should be easily available.

6.3 Case Studies of Software incidents

This section will now focus on case studies of popular security incidents in recent years. Analyzing these incidents will allow for further insight on possible risks this way of developing and distributing software has. It will show in real-world examples that next to affecting ways to detect and fix actual bugs in code, the way of developing and collaborating in open source projects is also susceptible to more indirect types of attacks, which play a smaller role in conventional software development, as trust plays an important role in OSS.

Furthermore, investigating these security incidents will help to understand common themes and causes and thus indicate which aspects need to be improved or handled with care when developing open source software in the future.

6.3.1 Example for Open Source

Heartbleed One of the largest recent security incidents in open source software was the Heartbleed bug. The buffer over-read bug in the OpenSSL library was discovered in 2014.

Exploiting the Heartbleed bug, an attacker is able to read information from memory that is adjacent to the information he is authorised to access [8].

The bug was introduced in the functionality of the library handling heartbeat messages. These are used in highly available systems, where a signal will be sent regularly in order to ensure that a component has not failed. Since this is possible both, from client to server, as well as from server to client, the bug does not only make servers vulnerable but also clients when they connect to a malicious server.

Because of the plethora of applications and servers using OpenSSL the bug had an immense impact. Just the two web servers nginx and Apache2, which both were vulnerable, made up 66 % of sites of the internet at the time of discovery. In addition to web servers a variety of programs were vulnerabl, such as mail-servers, VPNs and many other network applications. [22].

Since the vulnerability was introduced two years before its discovery, many services had already deployed vulnerable versions.

Sice the fixing of the bug only required simply checking that the given length of a request is correct, the patch fixing the bug was released on the same day Heartbleed was disclosed [22].

As a response to the incident, the library LibreSSL was created as a fork of OpenSSL by the OpenBSD team in order to modernise the codebase and combat OpenSSL's apparent software bloat [28]. In its first week since creation the project removed 90'000 lines of code. Since then, OpenSSL has been replaced by LibreSSL in several applications and operating systems, such as OpenBSD and Dragonfly BSD. Other projects temporarily switched to or allowed selecting LibreSSL, for example Alpine Linux, Voidlinux, Gentoo, and Python [3].

Log4Shell In 2021 a zero-day vulnerability was discovered in the logging library Log4j. When the library logged a string containing a LDAP-URL it would lookup the URL, receive and execute code from the specified LDAP-server. An attacker could exploit this by passing and LDAP-URL to a server that he controls, effectively allowing remote arbitrary code execution (ACE) [12]. ACEs are considered to be one of the most severe vulnerabilities. The case has similarities with the Heartbleed bug, as it is a library used in wide range of applications in the internet and at the same time (or as a result of) is not kept minimal. Whereas in Heartbleed the cause was more an unmaintainably large codebase (Software bloat), in Log4j it is more a problem of feature creep, as the execution of remote code on LDAP-servers is a very niche use case but prone to cause vulnerablities.

sudo In 2023 a vulnerability in sudo was discovered. Sudo is one of the most used commands in Linux environments used for running commands as superuser or as some specified user. The privilege escalation bug allowed an attacker to use the commands `sudo -e` or its alias `sudoedit` to edit files on the system as any user. Even though the vulnerability cannot be exploited remotely, and thus an attacker needs be authorised as a user in the vulnerable system [14], it still is considered a very severe vulnerability (rated 7.8 of 10 by the National Institute of Standards and Technology) [29], since sudo is used on many systems where many users have non-superuser access (as opposed to personal computers), and thus a user being able to gain root privileges is critical.

Notable about this vulnerability is, that the vulnerable code has been included in `sudo` for almost 10 years [4], damaging the reputation of open source’s advantage in security ”many eyes see more than one“. Similar to OpenSSL in Heartbleed and `Log4j` in `Log4Shell`, `sudo` has a very large code base [51].

NPM backdoors In the examples above, the security incidents were caused by traditional bugs, i.e. mistakes made while writing the program. Due to the collaborative nature of open source software, different attack vectors also arise in the organisation and distribution of the software. While traditional proprietary software usually is developed as contract work or within a company and then used solely within the company, open source software needs to be more open. This is usually achieved by hosting code on public version control systems (such as `git`) and / or publishing the releases in repositories, such as the Python Package Index or the `npm` registry. An attacker who manages to take control over a developer’s account is able to publish new versions of the packages which will be included in packages that depend on it. This has happened several times in both repositories mentioned.

In 2021 the two packages `coa` and `rc` were compromised by an unidentified attacker. Both were popular packages, the first being a command-line options parser and the second a simple configuration file loader. Together they had 22 million weekly downloads at that time [44]. Presumably they were targeted, as they are not standalone programs, but rather utilities which are intended to be included in many different programs, leading to a wider spread of the malware.

Protestware NPM Another case that highlights the potential security risks caused by the collaborative nature of open source software emerged in the context of the recent escalation of the Ukrainian-Russian conflict. Even though protestware (Malware used for political activism) has existed for a long time, its occurrences recently spiked. An impactful instance on open source software happened in spring of 2022 when the maintainer of the popular `node-ipc`, a package for inter-process communication for `Node.js`, which was included in a variety of other projects, added malicious code to the package which detected the IP of the device it is running on, and in case of a Russian or Belarusian one, would start overwriting files on the local machine [13].

The Open Source Initiative responded in a statement to this incident and in more generality to the “weaponization of open source” by including malware in projects, as harming “all of open source” [37].

Attack on Linux Kernel In November of 2022 a paper “On the feasibility of stealthily introducing vulnerabilities in open source software via hypocrite commits” was published by researches at the University of Minnesota. During their research they attempted to introduce use-after-free vulnerabilities into the Linux kernel through hypocrite commits to investigate the security of open source software [49]. After the publication and the malicious nature of the commits becoming known, the University was banned from further contributing to the kernel and all previously included commits were removed from the kernel and re-reviewed [17]. While many of the commits were found to not be problematic

or were removed in review [17], according to kernel maintainer Jiri Kosina, at least one malicious commit made it into a kernel release [26].

This case also shows the additional attack vectors open source is susceptible to compared to conventional closed source software. After the incident the authors of the paper, Quishi Wu and Kangjie Lu, suggested requiring contributors to the kernel to formally confirm that their commits are not made in bad faith and also to verify the identities of the contributors. In their response, the kernel team reaffirmed that the “Linux community is built upon trust” [49] and therefore do not want to assume bad intent from contributors [49].

Similarly to the node-ipc protestware case, this incident shows the importance of trust between collaborators within open source projects.

6.3.2 Example for Closed source

Microsoft Exchange In 2021 several critical vulnerabilities were discovered in Microsoft’s Exchange e-mail server. The vulnerability allowed an unprivileged attacker to enable remote code execution[9][10][11], resulting in the risk of whole servers being compromised. Due to Exchange being a popular business software, and the servers often having high privileges within companies, this vulnerability was of high interest as an entrance to attacks on a larger scale. The simplicity of using the vulnerability and the availability of exploits increased the severity further. The German Federal Office for Information Security warned that unpatched systems should be seen as compromised and warned over 9’000 companies that they are affected in Germany alone [6].

6.3.3 Lessons learned

Impactful software vulnerabilities in open source software tend to share two key characteristics. On the one hand, the affected projects tend to be depended on by a large number of large projects. This causes the dependency to become a single point of failure with possible catastrophic results. This was the case in all above mentioned security incidents caused by bugs in open source software, especially in Heartbleed and Log4Shell.

On the other hand the affected programs usually have a large code base with little aim to restrict keep the project minimal (e.g. due to software bloat or feature creep). This is especially dangerous when the development team is small and thus not able to appropriately maintain the code base anymore.

Besides the risks for unintended bugs to arise in open source software, the nature of how OSS is developed is fundamentally different from proprietary software. While conventional proprietary is usually developed by one company or strongly connected team, open source is developed in a way more decentralised and collaborative way. Even large projects often include foreign packages created and maintained by developers or teams which they know very little, have little connection to, and, compared to contract work in proprietary software, have almost no legal accountabilities from. All these factors highlight how important trust is in open source Software and its distribution and collaboration. Developers that include packages of others in their projects must not only be able to trust

their benevolence, but also be able to trust that their code follows appropriate security standards and they ensure the safety of their means of distribution of their code. Cases like the protestware in node-ipc harm the trust in the benevolence of popular open source packages, while cases like the backdoors in the NPM packages rc and coa damage the trust in the safety of the infrastructure used to distribute packages.

6.4 Improve security of OSS

As can be seen from the case studies, the use of open source introduces various types of attack vectors into a system. However, a strict security first approach can reduce the attack surface. In the following chapters, this work will first give a high-level overview of the different kinds of steps that need to be taken to create secure software. These steps help to minimise security problems in software development. After that, several tools are presented to give a more detailed introduction to some of the key steps.

6.4.1 Create secure software

Testing a piece of software for its overall security is a key part before deploying it in the wild. This applies for self created software but also for software written by others, as the safety of those cannot be guaranteed. This section will present a short summary of the most important steps according to the publication of NIST "Guidelines on Minimum Standards for Developer Verification of Software"[5].

Threat Modeling: This method is about capturing threats of the software and represent them in a more global context of the overall system. The goal is to find the parts of the software, where a security issue has the greatest impact on the system. These findings can then be reviewed manually.

Automated Testing: Automated testing should be part of any development circle of software. It is not only limited to verify the security but also functionality of software. The idea is to easily run some program which checks the software under test according to some rules. These tests can reach from small isolated and fast tests to huge complex tests which can take multiple hours to run through. Testing should be made as effortless as possible, to encourage developers to run tests as often as possible.

Code-Based, or Static, Analysis: This analysis step is about inspecting the program code with a scanning software. It is also called a static analysis because the dynamic program flow is mostly ignored and the system is more analysed without even running the program code. These scanners look for security issues that can be seen from a code inspection similar as a manual reviewer would do.

Review for Hardcoded Secrets: The idea is to scan the code for hard coded passwords or keys. Sometimes those are hidden but can be found with heuristic tools.

Run with Language-Provided Checks and Protection: NIST recommends running the code with Language-Provided Checks and Protection. Many programming languages or platforms to run software already have security mechanisms build into them. It is the responsibility of the developer or user to actually use them.

Black Box Test Cases: These test cases are for software, where the actual internal solution

is disregarded. The test cases are derived from the requirements specification that states what the software should do and should not do. For example it should be able to cope with a specific attack.

Code-Based Test Cases: Those tests are matched to the implementation of the software. It is recommended to start with black box and then move to those tests, as they might supply additional test cases that are not thought of before.

Historical Test Cases: The third kind of test cases are test cases that aim for a specific bug in the software. In the beginning it points out the bug, after the fixing it should show the absence.

Fuzzing: Fuzzing is a kind of testing where the input for the test is generated on the fly. The benefit is that the effort to test the software with a large number of inputs is very little in comparison to the test cases where the developer exactly specifies the input.

Web Application Scanning: This step aims for software that provides an web service. NIST basically mentions that some of the above checks should also done via the web interface of an application. For example testing the API with a fuzzing tool to generate random input values.

Check Included Software Components: The last method to verify the security of software is to check the included software components. Included software components can range from background database software to OSS libraries imported into program codes. Checking refers to running the important steps above with the included software as well to verify its integrity before using it.

The strategy mentioned above does not only aim at the creation process of software but also at the rest of the life cycle. During the phase of use the software has to be actively maintained. This includes running the above checks in a timely interval to continuously verifying the security of the software. Even if the software does not change, the included software might do. Therefore, also scanning the dependencies in a regular interval is a mandatory process.

At the end of the life cycle if the software is not longer needed, there are a few methods to consider. Some of the steps mentioned at the beginning of this chapter should be thought of in a reverse form. For example not only the software but also its dependencies have to be removed if they are no longer needed: those could be a running database on the system or some logging tools. It could have a negative impact to forget old and unmaintained dependencies on the system, as they may be the software with the vulnerabilities in the link. Also it might be a good idea to remove all the secrets and private keys that are no longer needed. This might not have an active security impact if it is not done but it can lead to additional confusion for the developers [5].

6.4.2 Example Tools

Code review is an important step on ensuring the quality and security of software. While this step oftentimes is still done manually, even in big projects, like it was the case in the Linux kernel for a long time [49], only relying on manual review can get overwhelming with fast paste development and a high frequency of merge requests to review.

This section will present two types of automated security checks, software component

analysis (SCA), a analysis on a higher level mainly based on metadata, and static application security testing (SAST), which goes more into depth but is computationally more costly to integrate into the systems development life cycle. For both an example tool will be shown.

6.4.2.1 Software Composition Analysis

As outlined in the chapter about risks when using OSS, managing the dependencies is a central key to a secure software. As it is not feasible to do this task manually, this chapter will introduce some tools to deal with this key part of maintaining software. This is also called Software Composition Analysis (SCA). Several tools will be introduced here. Software component analysis checks included dependencies against lists that store the versions of dependencies that contain known software vulnerabilities. It therefore does not analyse the code of the third party component or the project itself, but rather checks if this dependency is known to be insecure. It therefore can be applied very quickly and generally does not lead to false positives. For dependencies that are uncommon and thus not included in these lists, it will therefore, however, not prevent insecure software to be included [16][25]. Running such a tool is an important step to verify the security of some software as earlier introduced.

First, this chapter will look at the tool OWASP Dependency-Check. This tool is developed by the Open Worldwide Application Security Project (OWASP). It represents a "nonprofit foundation that works to improve the security of software" [33]. As this foundation claims to be as transparent and open as possible, it comes to no surprise that the produced tools by them are published as open source software.

The tool Dependency-Check is not the most advanced in comparison to others regarding additional feature implementations however it is an open source project and therefore is related to the topic. This tool has several versions depending on the use case. One of them is to run the tool from the Command Line Interface (CLI). It can be run on multiple types of source files of programs. It primarily supports C# and Java. However, support for other programming languages is available. The scanner searches the source code for its dependencies and compares them to the ones known to have vulnerabilities. The knowledge about vulnerabilities in specific versions of OSS libraries is public available online in the NVD CVE database which is hosted by the NIST [24].

The OWASP also created a plugin for the Azure DevOps Pipeline, so that it facilitates of adding the scanner for vulnerabilities to the CI / CD pipeline [34]. Since also an official docker image is available online this setup can be transported to literally every development pipeline on other DevOps platforms such as Gitlab or GitHub.

To sum up, this tool is a free little but nifty helper to defend against outdated dependencies. The automation makes it easily possible to integrate it in any software engineering workflow and therefore is highly recommend to be used. There are several commercial products with a similar workflow available. Since they mostly do the same, this work will not discuss them any further.

Second, there are several DevOps platforms that already have a SCA Scanner integrated. This chapter will have a look at the scanners of the platforms Gitlab or GitHub, which were already mentioned. These platforms are the key players in the code hosting market.

Therefore, most of the developers host their code on one of the two platforms.

GitLab is a DevOps platform that both has a free public cloud version but also offers a self-hosted solution. This way enterprises can host their own instance and keep the details of the development process hidden in their own network. Nevertheless, there is one official and multiple private instances that can be used by the public to host the development process for free. It is possible to integrate a scan step into the development pipeline. For example, when committing code to some branch, the dependencies get scanned and results will be displayed in the web interface of GitLab. In the context of open source, it is worth mentioning that there are two versions for self-hosting: one free and open source and an enterprise version [21].

GitHub is very similar to GitLab. There is also one public instance and an enterprise version that can be self-hosted. The SCA scanner in GitHub scans the code after each update and notifies the user if there are any issues. GitHub has a nice additional feature to automatically propose any fixes for the issue. If enabled, GitHub will create pull requests with fixed dependencies if any dependency has security vulnerabilities and should be updated or changed [31].

On a side note, if the development process is open-sourced, the scanner results are in fact always visible to the public as well. As explained earlier, this is a benefit for the user of this software to have the transparency but could also be used to search for exploits in the software.

To put this into a nutshell, there are several open-source tools to check dependencies and there are even platforms that take care of that for you as a developer. Utilizing such tools in the development process can extremely minimize the risk of vulnerable dependencies. As deploying is as easy as pressing a few buttons in the web interfaces of these code hosting platforms, there is no reasonable decision to not use these kind of SCA scanners.

6.4.2.2 Static Application Security Testing

In contrast to software component analysis, the static application security testing (SAST) analyzes the actual code instead of just checking the included dependencies against a list of known vulnerable versions like SCA. In this process, patterns that could cause security vulnerabilities are recognized and reported to the developer for review [16].

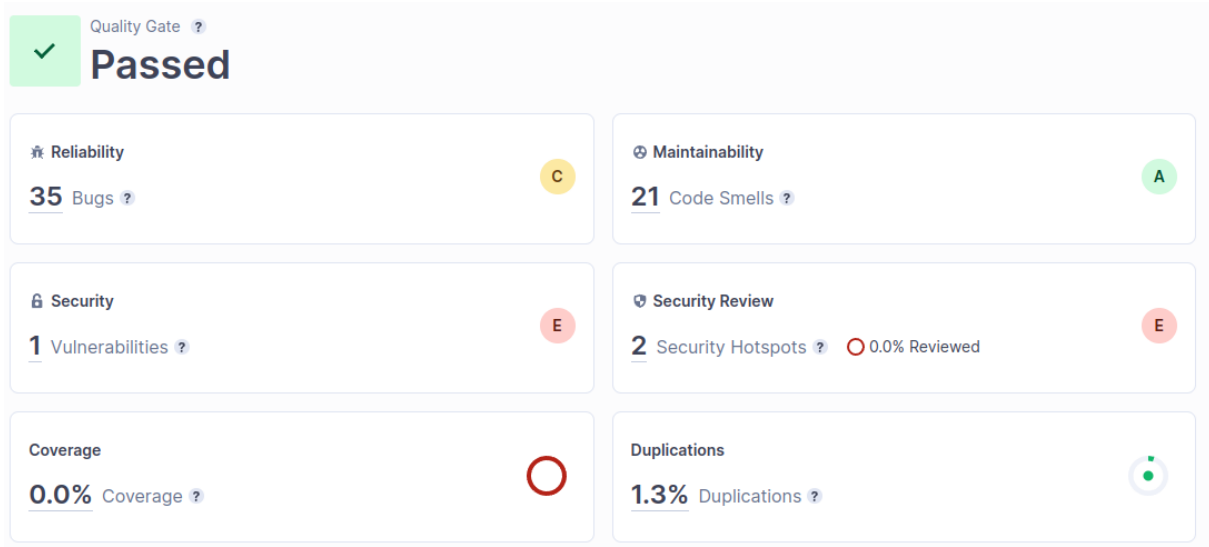


Figure 6.5: Screenshot of SonarQube’s dashboard showing security vulnerabilities

A popular tool used for static application security testing is the open source software SonarQube. It is used by over 400’000 companies like Microsoft, IBM, and Siemens [40]. It offers a self-hosted and cloud-version (SonarCloud) which has free plans for open source projects [39]. Many of the projects have public dashboards where anyone can look at the analysis of the code. Examples are the IDE VS Code [43], the web application framework Apache Struts 2 [41], or the Brave Browser [42]. Due to open nature of the SAST tool, on the one hand open source projects, which often do not have a lot of financial resources to spend on these kind of tools, can profit from it, helping the project to spot vulnerabilities and improve the security of their code, and on the other hand, developers that want to include or build upon software that uses SonarQube can more easily evaluate the current security state of the code, helping to avoid including external security vulnerabilities into their projects.

Here the impact open source has on security is visible by on the one hand making analysis tools like this more accessible to also financially weak projects and developers and on the other hand facilitating evaluation of dependencies, which impacts open source as well as proprietary software.

6.5 Summary

In conclusion, this study showed that there are benefits and risks of open source products in terms of cybersecurity. It is important to keep in mind that not every project will automatically gain these benefits from open sourcing, but will also depend on many external factors. This can also be seen in the cases studied where experienced developers of large open source code bases still had security problems in their software.

Not only does open source affect how well security incidents can be spotted and fixed but because open source changes software development in a fundamental way, new attack

vectors are opened, such as attacking the distribution of the software, introducing malicious code by exploiting insecure accounts on collaboration platforms or even abusing the required trust in open source by deliberately including malicious code in one's own software. The possible reusability of open source code in other programs, on the one hand causes risks when many projects built upon the same vulnerable code base, as seen in Heartbleed, Log4Shell, `coa`, `rc` and `node-ipc`, but on the other hand also enable developers to use well-built and tested software. One could only imagine if every small project would need to build its own cryptographic libraries.

Today, however, it is almost impossible to develop software without relying on other developers through open source libraries. It is therefore likely that there will be further growth in the use of open source software in the future.

From user's as well as developer's view, open source increases transparency. For example by allowing to more easily evaluate the security of used software and collectively or individually ensure that software does not have functions against the user's interest.

The discussion in one of the chapters has also illustrated the benefits and risks of the offensive side of open source attack tools in cybersecurity. This simply reflects the question of whether all information and tools should be available online. However, several examples over time have shown that all stakeholders benefit from freedom of information and additional transparency.

Ultimately, the question is not how to avoid or remove open source software because of its potential lack of security, but how to deal with the risks. As shown in the previous chapter, there are several ways to improve security when using public code. The key is to recognise that OSS is not a universal remedy for securing software, but a way of helping the user and developer along the way. But in the end, the user still has to take care of the software himself.

Bibliography

- [1] Aiko Pras, Anna Sperotto, Giovane C. M. Moura, Idilio Drago, Rafael Barbosa, Ramin Sadre, Ricardo Schmidt, Rick Hofstede Attacks by 'Anonymous' WikiLeaks Proponents not Anonymous. , <https://ris.utwente.nl/ws/portalfiles/portal/5095225/2010-12-CTIT-TR.pdf>
- [2] Andrea Peterson The Sony Pictures hack, explained. The Washington Post., <https://www.washingtonpost.com/news/the-switch/wp/2014/12/18/the-sony-pictures-hack-explained/>
- [3] Ars Technica. (2014). OpenSSL code beyond repair, claims creator of “LibreSSL” fork. Retrieved 2023-05-31 from <https://arstechnica.com/information-technology/2014/04/openssl-code-beyond-repair-claims-creator-of-libressl-fork/>
- [4] Beyond Trust. (2021). Understanding Sudo Vulnerability CVE-2021-3156 and How Privilege Management for Unix & Linux Can Protect Your Enterprise. Retrieved 2023-04-18 from <https://www.beyondtrust.com/blog/entry/understanding-sudo-vulnerability-cve-2021-3156-and-how-privilege-management-for-unix-linux-can-protect-your-enterprise>
- [5] Black, P., Guttman, B. & Okun, V. Guidelines on Minimum Standards for Developer Verification of Software. (National Institute of Standards and Technology, 2021), <https://nvlpubs.nist.gov/nistpubs/ir/2021/NIST.IR.8397.pdf>
- [6] Bundesamt für Sicherheit in der Informationstechnik. (2021). BSI warnt: Kritische Schwachstellen in Exchange-Servern (archiviert). Retrieved 2023-04-18 from https://www.bsi.bund.de/DE/Service-Navi/Presse/Pressemitteilungen/Presse2021/210305_Exchange-Schwachstelle.html
- [7] Burgess, M., Newman, L. H. (2023, January 24). The Unrelenting Menace of the LockBit Ransomware Gang. WIRED UK. Retrieved from <https://www.wired.co.uk/article/lockbit-ransomware-attacks>
- [8] CVE. (2014). CVE-2014-0160. Retrieved 2023-03-30 from <https://www.cve.org/CVERecord?id=CVE-2014-0160>
- [9] CVE. (2021). CVE-2021-28480. Retrieved 2023-04-18 from <https://www.cve.org/CVERecord?id=CVE-2021-28480>
- [10] CVE. (2021). CVE-2021-28481. Retrieved 2023-04-18 from <https://www.cve.org/CVERecord?id=CVE-2021-28481>

- [11] CVE. (2021). CVE-2021-28482. Retrieved 2023-04-18 from <https://www.cve.org/CVERecord?id=CVE-2021-28482>
- [12] CVE. (2021). CVE-2021-44228. Retrieved 2023-03-29 from <https://www.cve.org/CVERecord?id=CVE-2021-44228>
- [13] CVE. (2022). CVE-2022-23812. Retrieved 2023-04-19 from <https://www.cve.org/CVERecord?id=CVE-2022-23812>
- [14] CVE. (2023). CVE-2023-22809. Retrieved 2023-03-31 from <https://www.cve.org/CVERecord?id=CVE-2023-22809>
- [15] Chen Han, Rituja Dongre. (2014). Q&A. What Motivates Cyber-Attackers? Technology Innovation Management Review, 4(10). Retrieved from <https://www.timreview.ca/article/838>
- [16] Cloud Defense. The Differences Between SCA, SAST and DAST. Retrieved 2023-05-18 from <https://www.clouddefense.ai/blog/the-differences-between-sca-sast-and-dast>
- [17] Cook., C. (2021). Report on University of Minnesota Breach-of-Trust Incident. Retrieved 2023-05-18 from <https://lore.kernel.org/lkml/202105051005.49BFABCE@keescook/>
- [18] Das, A. (2021, July 18). Is Open-Source Software Secure? It's FOSS News. Retrieved from <https://news.itsfoss.com/open-source-software-security/>
- [19] Diaz Ferreyra, N., Imine, A., Vidoni, M. & Scandariato, R. nicoediaz/PRIVACY-SCPS: Initial Release. (Zenodo,2023), <https://arxiv.org/pdf/2303.01822v1.pdf>
- [20] Free Software Foundation. (2022). What is Free Software. Retrieved 2023-03-29 from <https://www.gnu.org/philosophy/free-sw.html>
- [21] GitLab B.V. Dependency Scanning | GitLab. , https://docs.gitlab.com/ee/user/application_security/dependency_scanning/
- [22] Heartbleed. (2020). Heartbleed. Retrieved 2023-03-30 from <https://heartbleed.com/>
- [23] Huddleston, T. (2022, March 25). What is Anonymous? How the infamous 'hactivist' group went from 4chan trolling to launching cyberattacks on Russia. CNBC.Retrieved from <https://www.cnbc.com/2022/03/25/what-is-anonymous-the-group-went-from-4chan-to-cyberattacks-on-russia.html>.
- [24] Jeremy Long dependency-check - How does dependency-check work. , <http://jerymylong.github.io/DependencyCheck/general/internals.html>
- [25] Katz, E. Top 8 Software Composition Analysis (SCA) Tools for 2023 - Spectral. (2023), <https://spectralops.io/blog/top-8-software-composition-analysis-sca-tools-for-2023/>

- [26] Kosina, J. (2021). Re: [PATCH 000/190] Reversion of all of the umn.edu commits. Retrieved 2023-05-18 from <https://lore.kernel.org/lkml/YIBBt6ypFtT+i994@pendragon.ideasonboard.com/T/#ma4bc909958aea19d8cc9e10a8182e1b3c06e3056>
- [27] Lakunishok, B. (2022, May 12). Can Open Source In Cybersecurity Be A Game Changer? Forbes. Retrieved from <https://www.forbes.com/sites/forbestechcouncil/2022/05/12/can-open-source-in-cybersecurity-be-a-game-changer/>
- [28] LibreSSL. (2023). LibreSSL. Retrieved 2023-03-30 from <https://www.libressl.org/>
- [29] National Institute of Standards and Technology. (2023). CVE-2023-22809 Detail. Retrieved 2023-04-18 from <https://nvd.nist.gov/vuln/detail/CVE-2023-22809>
- [30] National Vulnerability Database. (2023). CVE-2023-22809 Detail. Retrieved 2023-05-31 from <https://nvd.nist.gov/vuln/detail/CVE-2023-22809>
- [31] O'shea, B. SCA vs SAST: what are they and which one is right for you?. *GitHub.*, <https://github.blog/2022-09-09-sca-vs-sast-what-are-the-y-and-which-one-is-right-for-you/>
- [32] OWASP A06 Vulnerable and Outdated Components - OWASP Top 10:2021. , https://owasp.org/Top10/A06_2021-Vulnerable_and_Outdated_Components/
- [33] OWASP About the OWASP Foundation | OWASP Foundation. , <https://owasp.org/about/>
- [34] OWASP Dependency Check OWASP Dependency Check - Visual Studio Marketplace. , <https://marketplace.visualstudio.com/items?itemName=dependency-check.dependencycheck>
- [35] Okeke, F. Top 10 open-source security and operational risks of 2023. *TechRepublic.*, <https://www.techrepublic.com/article/top-open-source-security-risks/>
- [36] Open Source Initiative. (2007). The Open Source Definition. Retrieved 2023-03-29 from <https://opensource.org/osd/>
- [37] Open Source Initiative. (2022). Open source 'protestware' harms Open Source. Retrieved 2023-04-19 from <https://blog.opensource.org/open-source-protestware-harms-open-source>
- [38] Ron Rymon. (2021). 3 Reasons Why Open Source Software is More Secure than Commercial Software. Retrieved from <https://www.mend.io/resources/blog/3-reasons-why-open-source-is-safer-than-commercial-software/>
- [39] Sonar. As a service. Retrieved 2023-05-18 from <https://www.sonarsource.com/products/sonarcloud/>
- [40] Sonar. Customers & Organizations using Sonar. Retrieved 2023-05-18 from <https://www.sonarsource.com/company/customers/>

- [41] Sonarcloud. Struts 2. Retrieved 2023-05-18 from https://sonarcloud.io/project/overview?id=apache_struts
- [42] Sonarcloud. brave-core. Retrieved 2023-05-18 from https://sonarcloud.io/project/overview?id=brave_brave-core
- [43] Sonarcloud. vscode-python. Retrieved 2023-05-18 from https://sonarcloud.io/project/overview?id=microsoft_vscode-python
- [44] The Hacker News. (2021). Two NPM Packages With 22 Million Weekly Downloads Found Backdoored. Retrieved 2023-04-05 from <https://thehackernews.com/2021/11/two-npm-packages-with-22-million-weekly.html>
- [45] The Hacker News. (2023). Researches Hijack Popular NPM Package with Millions of Downloads. Retrieved 2023-04-05 from <https://thehackernews.com/2021/11/two-npm-packages-with-22-million-weekly.html>
- [46] The Open Group. The UNIX Standard. Retrieved 2023-05-31 from <https://www.opengroup.org/membership/forums/platform/unix>
- [47] Tim Maurer & Robert Morgus. Compilation of Existing Cybersecurity and Information Security Related Definitions. (New America Foundation), <https://d1y8sb8igg2f8e.cloudfront.net/documents/compilation-of-existing-cybersecurity-and-information-security-related-definitions.pdf>
- [48] Tim Maurer, Robert Morgus. (2014). Compilation of Existing Cybersecurity and Information Security Related Definitions. Retrieved from New America Foundation website: <https://d1y8sb8igg2f8e.cloudfront.net/documents/compilation-of-existing-cybersecurity-and-information-security-related-definitions.pdf>
- [49] Wu, Q., & Lu, K. (2021). On the feasibility of stealthily introducing vulnerabilities in open-source software via hypocrite commits. Proc. Oakland.
- [50] g0tmilk. (2022). What is Kali Linux? | Kali Linux Documentation. Retrieved from <https://www.kali.org/docs/introduction/what-is-kali-linux/>
- [51] sudo. Repository. Retrieved 2023-05-31 from <https://www.sudo.ws/repos/sudo>
- [52] vx-underground. (2023). Retrieved from <https://www.vx-underground.org/>

Chapter 9

ChatGPT in Network Security - An Overview of its Capabilities for Good and Bad Use Cases

Ahmad Eynawi

Contents

9.1	Abstract	167
9.2	Introduction	168
9.2.1	Background	168
9.2.2	Capabilities of ChatGPT	169
9.3	ChatGPT in Cybersecurity	170
9.3.1	Good Use Cases	171
9.3.2	Bad Use Cases	178
9.4	Applications of ChatGPT in Network Security - Malware Development - Practical Example	183
9.5	Applications of ChatGPT in Network Security - Anti-Malware Devel- opment - Practical Example	187
9.6	Limitations	189
9.7	Ethical Aspects and Solutions so far	193
9.8	Summary and Conclusion	193

9.1 Abstract

Almost no technology has recently attracted as much attention as OpenAI's ChatGPT since it was launched in November 2022. ChatGPT is a large language model trained by OpenAI, an American AI research company. It has numerous applications in natural language processing, including in the field of network security and cybersecurity in general.

Over the last few months since its emergence, discussions arose about the impact this technology will have on the field of cybersecurity with its both offensive and defensive variant. Several applications of ChatGPT have been shown in the field of cybersecurity such as in the areas of intrusion detection, threat intelligence, malware analysis, security consulting, etc. For example, ChatGPT can help organizations improve their security posture and better protect their systems from a wide range of threats. Furthermore, ChatGPT is a valuable education aid since it is really professional at providing compact and easily comprehensible answers to almost any question one might think of. It can, for example, be used to help users set up secure passwords or to provide guidance on how to avoid common threats like social engineering attacks, improving the cybersecurity awareness of the general public.

Yet, since ChatGPT is still a quite new and rapidly evolving technology, its impact on cybersecurity might be or become much bigger than one might expect today (at the end, this is why groundbreaking technologies are called "groundbreaking").

In this report, we will shed light on some applications of ChatGPT in the area of defensive as well as offensive cybersecurity. We will also introduce two practical examples that show some strengths and weaknesses, bright and dark sides as well as some limitations of this double-edged sword.

9.2 Introduction

Since its release in November 2022, specialists have been studying the impacts ChatGPT will have on different areas of study and research. Being a conversational system that has been trained on a huge and diverse corpus of data from the internet, the use cases of ChatGPT are almost as broad as the use cases of the internet itself. We devoted this work to study the applications of ChatGPT in the area of cybersecurity in particular. In this introductory chapter, we will provide an overview of the most important capabilities and characteristics of ChatGPT. After that, we will elaborate on some good and bad use cases that demonstrate the dual nature of this technology.

9.2.1 Background

ChatGPT is an intelligent chatbot developed by OpenAI [28] and launched on November 30, 2022. It is built on top of OpenAI's Generative Pre-trained Transformers GPT-3.5 and GPT-4 and has been fine-tuned for generating natural language answers. Generative Pre-trained Transformers (GPT) are large language models that use supervised and reinforcement learning techniques to produce human-like text by predicting the most probable next token in a sequence of tokens based on previous tokens (generative pre-training).

GPT-3.5 and GPT-4 are the successors of GPT-3, the third generation of OpenAI's GPT large language models. GPT-3 was released by OpenAI in 2020 and was trained on a gigantic corpus of online texts. The architecture of GPT-3 is a decoder-only transformer [12] network with a size of 175 billion learnable parameters that are adjusted during the training process to best match and continue the prompt.

In March 2022, OpenAI launched GPT-3.5, a fine-tuned version of the GPT-3 large language model. GPT-3.5 is based on GPT-3 but works within specific policies of human values and was optimized to provide more accurate and coherent responses. It was trained on the same datasets of GPT-3 but with more learnable parameters (355 billion) and additional fine tuning process that adds a concept called *Reinforcement Learning with Human Feedback (RLHF)* to the GPT-3 model [11]. RLHF is a subfield of artificial intelligence that focuses on using human feedback to improve machine learning algorithms. In RLHF, the human provides feedback to the machine learning algorithm, which is then used to adjust the model's behavior. This approach is used to address the limitations of supervised and unsupervised learning, where the machine learning algorithms have limited ability to learn from just labeled or unlabeled data. Using RLHF techniques, the model can incorporate human expertise and knowledge and improve its performance and abilities to solve complex tasks.

GPT-3.5 was trained on online datasets available up until September 2021. Hence, it does not have knowledge of events or developments that occurred beyond this date (unless they are explicitly shared with the bot during conversations with humans).

In addition to GPT which is designed for understanding and generating human-like natural language, ChatGPT also employs two other model families which serve slightly different

purposes. One of those families is *Codex*, which is specialized on translating human-like natural language into computer programming languages, effectively generating source code based on prompt instructions. The other model family is *Embeddings*, which is focused on specialized functionalities such as text search, similarity and code search, enabling more efficient information retrieval and processing in various contexts [1].

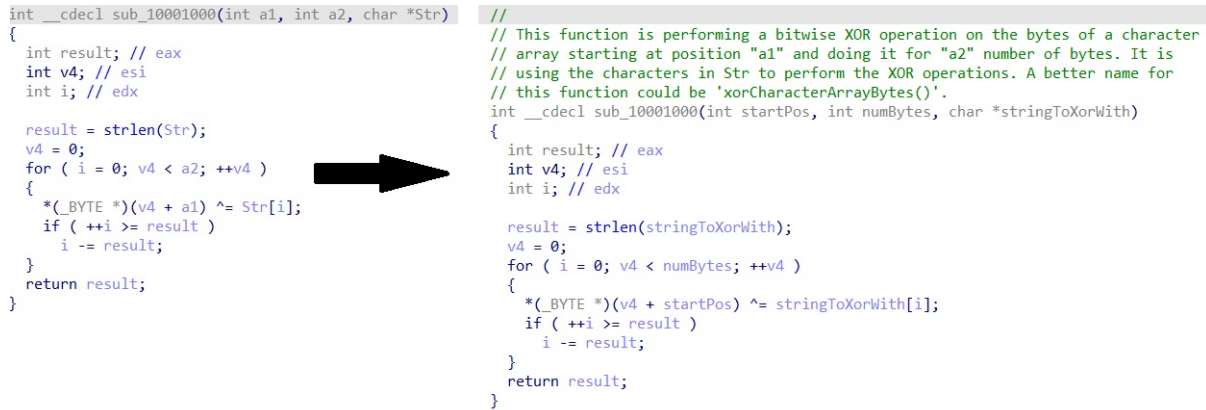
At this point, it should be noted that the statements and results mentioned in this report refer to ChatGPT with GPT-3.5 as underlying language model.

9.2.2 Capabilities of ChatGPT

Even though there have been many language models before the emergence of ChatGPT, all of them had their weaknesses and drawbacks which prevented them from gaining the popularity that ChatGPT has gained right after its emergence in November 2022. ChatGPT, however, comes with a lot of strengths that made it to one of the groundbreaking inventions in the AI world. To list some of its features, ChatGPT is capable of [10]:

- making coherent conversations with human-like text and engaging in natural language dialog with humans
- giving appropriate context-specific and user-tailored personalized responses in conversations, maintaining a coherent conversation with the user
- answering questions on a wide range of topics and giving precise, compact and easy-to-understand answers
- writing and explaining code
- translating code between programming languages
- translating text between languages
- classifying text based on its content
- completing text based on a given prompt and summarizing large amounts of text
- writing business letters, emails, reports and and blog posts
- explaining scientific technical concepts in an easily understandable way
- imitating a certain speaking or programming style

With respect to its coding capabilities, figure 9.1 shows ChatGPT explaining the code of a C function and suggesting a more meaningful name for the function [22].



```
int __cdecl sub_10001000(int a1, int a2, char *Str)
{
    int result; // eax
    int v4; // esi
    int i; // edx

    result = strlen(Str);
    v4 = 0;
    for ( i = 0; v4 < a2; ++v4 )
    {
        *(_BYTE *)(v4 + a1) ^= Str[i];
        if ( ++i >= result )
            i -= result;
    }
    return result;
}

// This function is performing a bitwise XOR operation on the bytes of a character
// array starting at position "a1" and doing it for "a2" number of bytes. It is
// using the characters in Str to perform the XOR operations. A better name for
// this function could be 'xorCharacterArrayBytes()'.
int __cdecl sub_10001000(int startPos, int numBytes, char *stringToXorWith)
{
    int result; // eax
    int v4; // esi
    int i; // edx

    result = strlen(stringToXorWith);
    v4 = 0;
    for ( i = 0; v4 < numBytes; ++v4 )
    {
        *(_BYTE *)(v4 + startPos) ^= stringToXorWith[i];
        if ( ++i >= result )
            i -= result;
    }
    return result;
}
```

Figure 9.1: ChatGPT explaining a function

9.3 ChatGPT in Cybersecurity

Since ChatGPT was launched in November 2022, there have been a lot of discussions about the impact it is going to have on the field of cybersecurity. It has received a lot of attention from the technology industry and the cybersecurity industry in particular. Even though, at the end, ChatGPT is first and foremost a language processing model that depends on pre-existent data to learn and develop, its ability to provide detailed yet precise answers across many domains of knowledge led many people to see it as a game changer for all fields including cybersecurity.

On the one hand, having the ability to write and analyze code makes ChatGPT a powerful tool for both blue as well red teamers. Because, at the end, both malware and anti-malware are programs developed with the aid of programming languages such as Python or JavaScript. So, the “toolkit” used by white- and black-hat hackers is essentially the same. What differs is only the “intention” and the purpose for which these tools are eventually utilized.

Furthermore, being able to engage in conversations and write coherent and convincing-sounding texts again shows the dual nature of this technology. On the one hand, this can help educate people about cybersecurity and raise the cybersecurity awareness in general. But on the other hand, this capability can also be misused by red teamers to launch powerful social engineering attacks such as pre-texting or spear phishing.

Therefore, like almost any technology, ChatGPT also has a shadow and a bright side [9]. In this section, we are going to discuss some selected good and bad use cases of this dual-use technology to demonstrate its dual nature and, on the one hand, show how ChatGPT can be a considerable aid for people interested in learning (more) about cybersecurity and, on the other hand, show why it is important to put more effort into the content policies of ChatGPT to improve its safe use by maximizing its benefit for non-malicious actors and minimizing it for malicious actors.

9.3.1 Good Use Cases

Generally, our experiments with ChatGPT have shown that the bot is rather not restricted in answering questions that can help blue teamers or even non-specialists defend against cyberattacks and improve their knowledge about cybersecurity. While non-specialists might want to use ChatGPT to easily improve their knowledge about cybersecurity without having to go through a plethora of books or articles on the internet, cybersecurity specialists may want to get rid of some routine tasks that do not necessarily require human intelligence and can be handed over to trained AI. The following sections list some of the most relevant good use cases of ChatGPT in the field of cybersecurity.

9.3.1.1 Security Consulting and Improving Cybersecurity Awareness

Since ChatGPT was trained on a huge text corpus from the internet, it knows a lot about what has been written online about cybersecurity, whether it is code or normal text. Therefore, it can provide compact and easily comprehensible answers to almost all topics related to cybersecurity. For example, we asked ChatGPT to provide us with some advice on how to protect ourselves from cyberattacks and got the following answer:

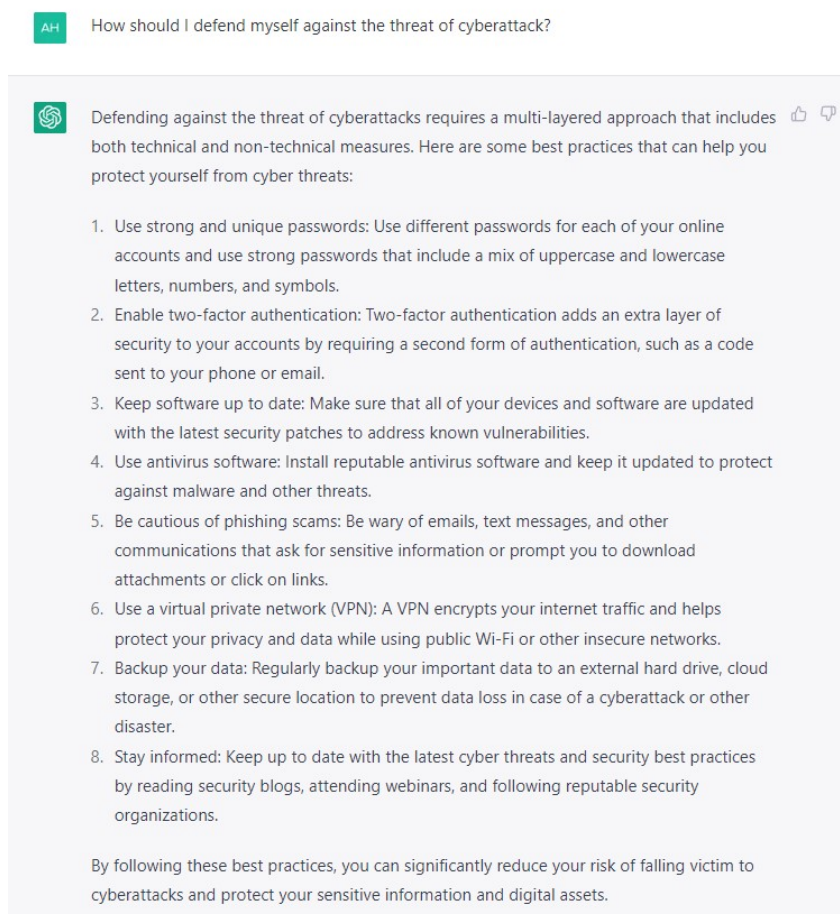


Figure 9.2: ChatGPT, how should I defend myself against cyberattacks?

Apart from being coherent and easy-to-understand, the answer given by the chatbot in fact includes the first tips one will very likely hear when talking about cybersecurity awareness. It is also worth mentioning that ChatGPT can provide even more tips (such as being aware of public Wi-Fis, encrypting sensitive data, implementing a firewall, etc.) for enhancing cybersecurity when it is explicitly asked to do so (for example, by asking the follow-up questions: *Can you give me more tips for enhancing my cybersecurity?*).

Another example shows a ChatGPT's answer to a question that is very commonly asked by security-aware business executives who want to develop an effective cybersecurity strategy for their companies:

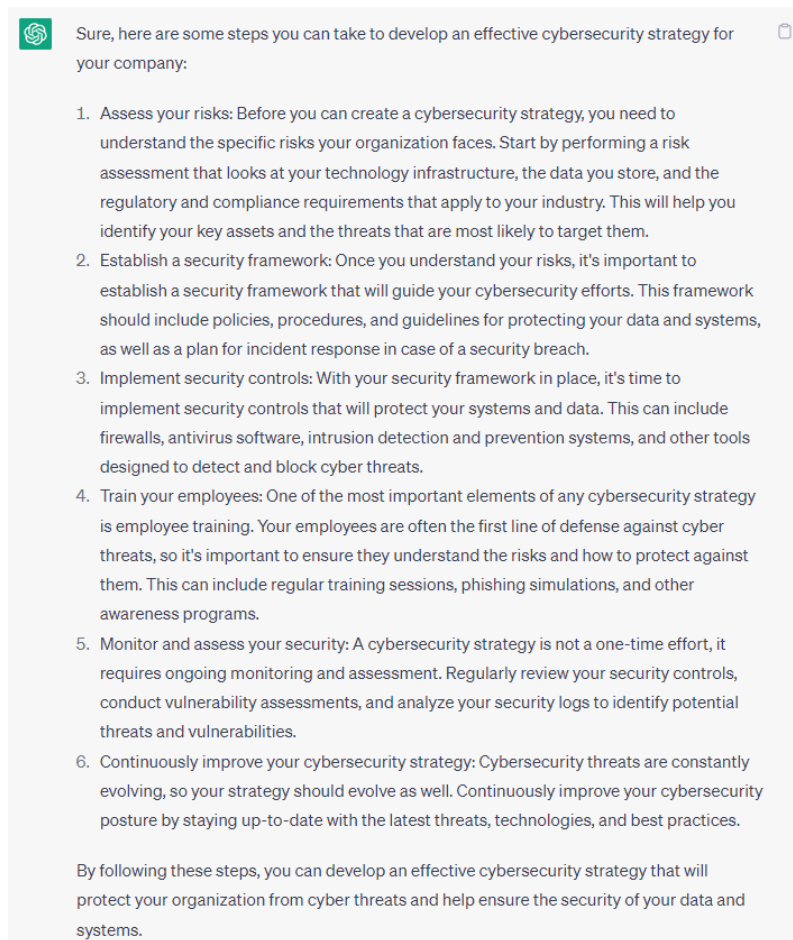


Figure 9.3: ChatGPT providing a Cybersecurity Strategy

In contrast to the question asked in figure 9.2, ChatGPT is also capable of generating satisfying answers to quite specific questions that might require some research effort to be answered by general-purpose non-conversational search engines. Based on the description of a specific real-world computer problem, we asked ChatGPT to provide us with possible reasons for a text program to start randomly generating characters without any user interaction, and we got the following answer:

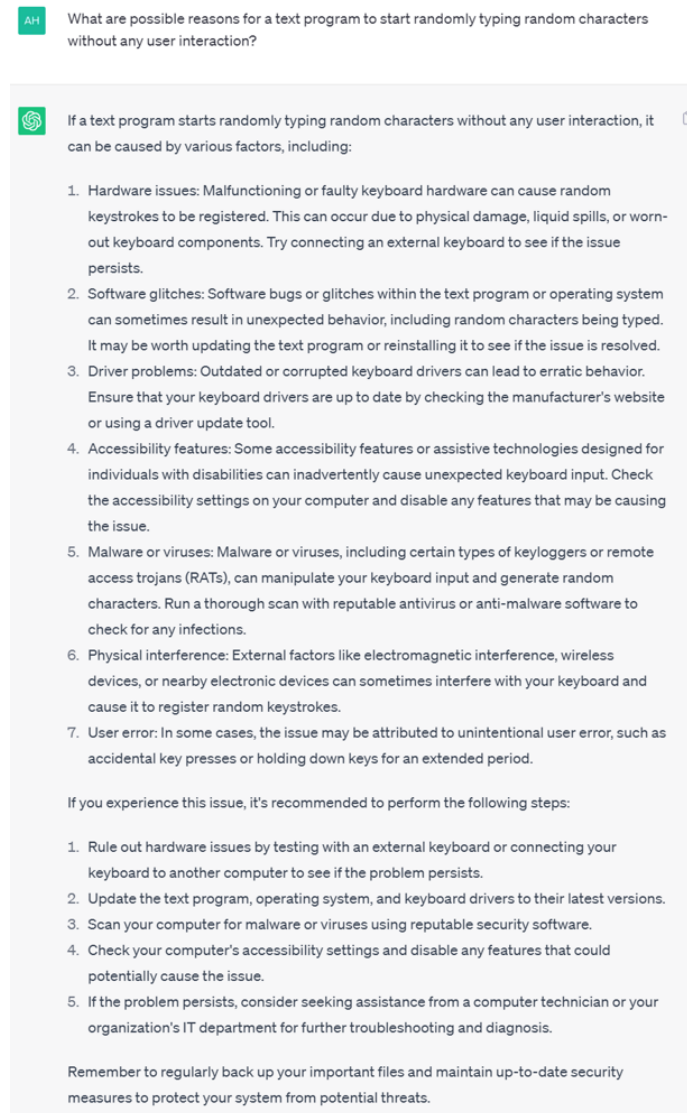


Figure 9.4: Possible reasons for keyboard hallucination

In bullet point 5, the answer contains a hint at the potential existence of a keylogger attack [8]. Hence, it seems natural to further ask GPT how keyloggers can be detected and how to recover from such types of attacks, as the following figure shows:

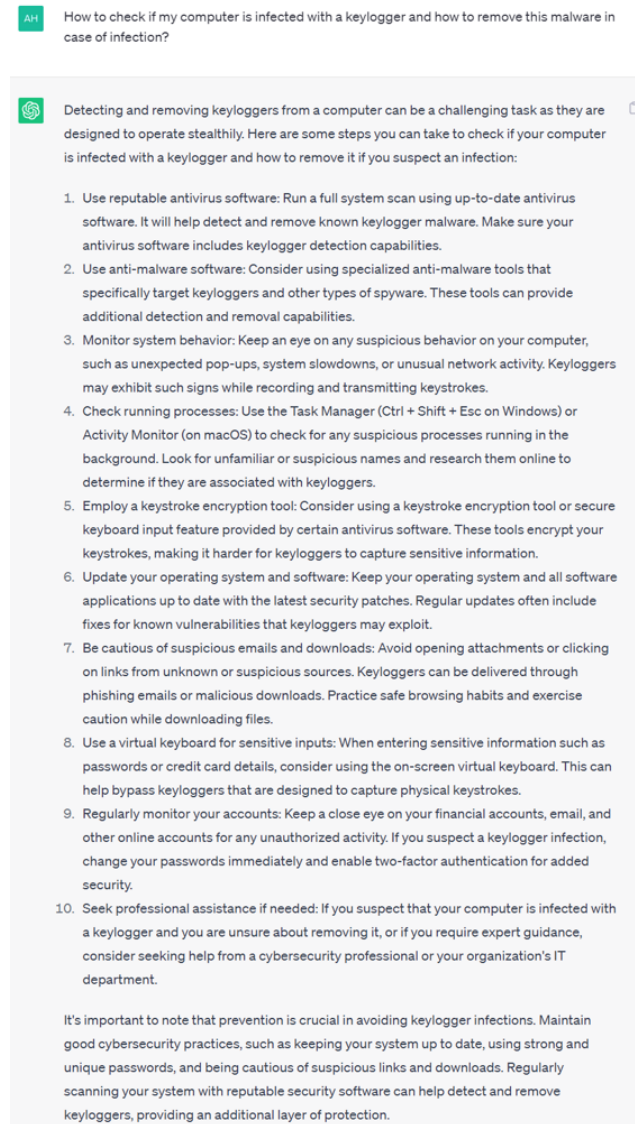


Figure 9.5: Detecting and recovering from keylogger attacks

Even though all these tips can be found somewhere in the internet, we can admit that getting such a quite accurate diagnosis without having to go through multiple web pages is a strength search engines like Google have not been able to attain since decades. Moreover, the fact that ChatGPT is a context-sensitive chatbot that can provide context-specific and personalized responses (or sequences of correlated responses) often leads to more satisfying and helpful results than those from general-purpose search engines that do not achieve the same level of context-sensitivity.

Those examples show that ChatGPT's ability to adeptly answer specific questions with highly relevant information, boiled down to a few paragraphs or bullet points, means that ChatGPT is often a faster way to learn about a new topic than combing through the web with search engines like Google. As a chatbot that can understand natural language, ChatGPT undoubtedly has the potential to improve user's understanding of cybersecurity and help them gain a quick and compact overview about security topics, reducing the knowledge gap for those who need or want to be aware of this area.

9.3.1.2 Vulnerability Analysis

In fact, ChatGPT sometimes identifies security weaknesses in code and provides suggestions to fix them, becoming a considerable aid for cybersecurity specialists who strive to analyse software vulnerabilities and remove them.

For example, the bot is capable of deliberately generating code that is vulnerable to certain types of attacks, such as buffer overflow attacks. However, at the same time, it can elegantly tell how to fix vulnerabilities discovered in code analysis such as buffer overflow vulnerabilities, as shown in figures 9.6 and 9.7:

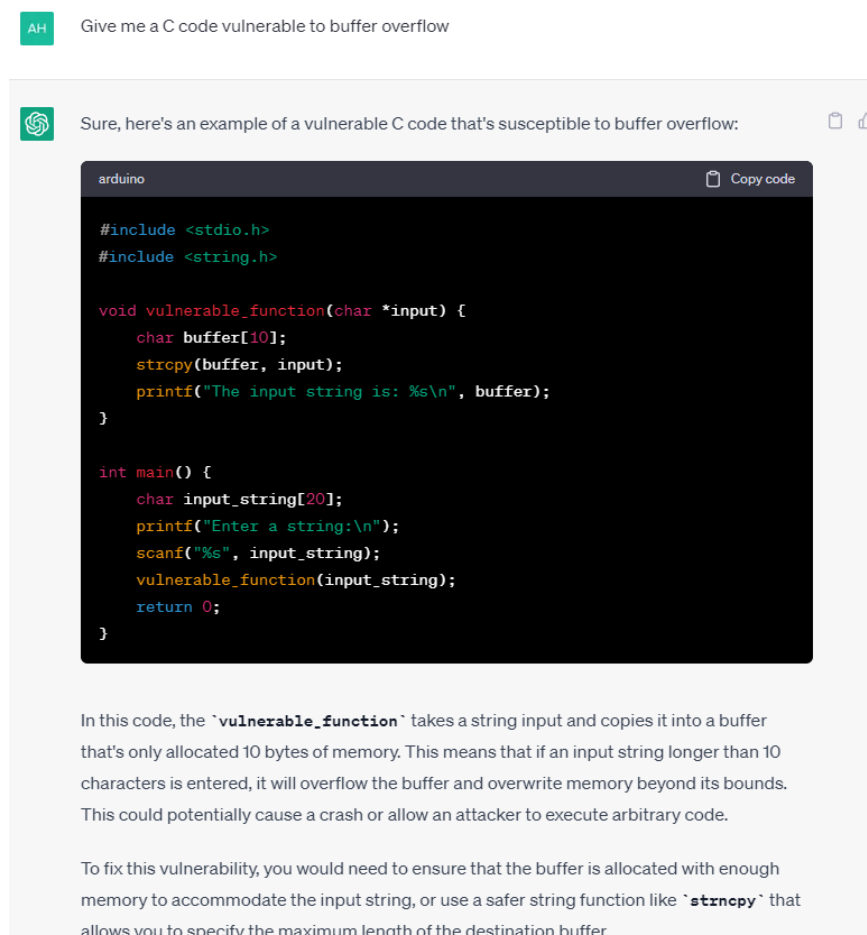


Figure 9.6: Generating C code vulnerable to buffer overflow

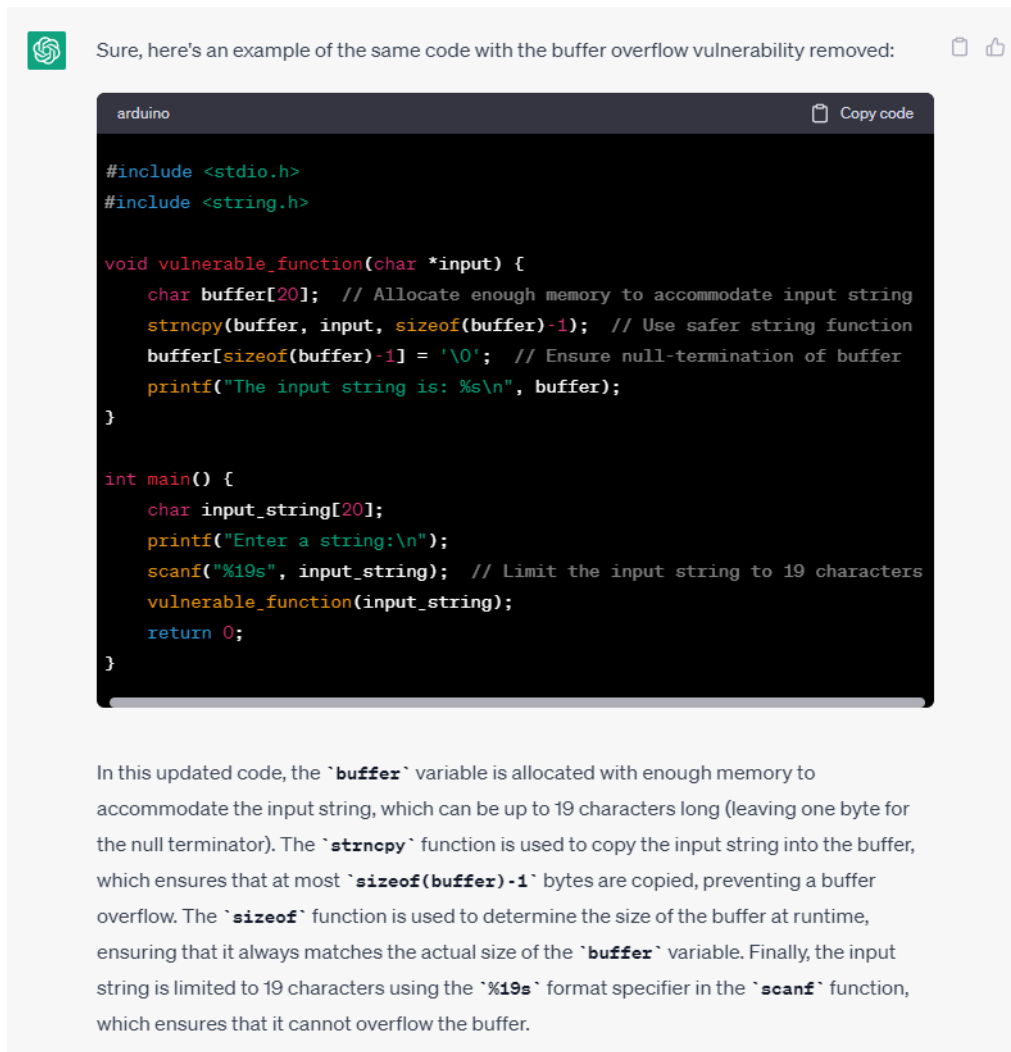


Figure 9.7: Removing buffer overflow vulnerability from C code

This example clearly shows the dual nature of this system. While it can be used by red teamers to learn about code vulnerabilities with the intention to exploit them, it can also be used by blue teamers to detect and fix vulnerabilities as part of a vulnerability analysis that is normally performed by security analysts or penetration testers (in open-box pen tests). In this sense, ChatGPT can be compared to other security-related tools, such as Cobalt Strike or Metasploit, which can be used for both defensive and offensive purposes alike. These tools are useful for legitimate penetration testing and simulating potential adversaries, but they are also used by black-hat hackers to help them break into the computer systems of their victims and engage in malicious activities.

At this point, it might be interesting to know that the recent update to GitHub Copilot¹ introduces an AI-driven vulnerability filtering system that enhances the security of its code suggestions. By detecting and preventing insecure code patterns such as hard coded credentials or SQL injections in real-time, this feature assists programmers in avoiding common coding mistakes, allowing them to develop more secure applications [1].

¹<https://github.com/features/copilot>

9.3.1.3 Automating many areas of cybersecurity

Even though automation has been a buzzword for several years now, many routine tasks in the field of cybersecurity are still done manually and there is still a lot of space and potential for automation in security event and information management. Although ChatGPT cannot be considered a standalone solution to automating security incident analysis, it has been proven to provide valuable aid in this field.

For example, ChatGPT is capable of generating so-called Sigma detection rules to identify cybersecurity anomalies [15]. The following YAML code shows a sigma rule written by ChatGPT to automatically detect Brute Force attacks:

```
title: Brute Force Login Attempts
description: Detects multiple failed login attempts that may
indicate a brute force attack
logsource:
  category: authentication
  product: windows
detection:
  selection:
    - event.action: "authentication failure"
  condition: >
    count_unique(event.target_user_name) > 5
fields:
  - "event.target_user_name"
  - "event.target_domain_name"
  - "event.source_address"
  - "event.target_server"
output:
  title: "Brute Force Login Attempt Detected"
  description: "Detected multiple failed login attempts
that may indicate a brute force attack"
  tags: [authentication, security, brute_force]
```

Moreover, there have been some use cases where ChatGPT was able to yield a useful story about a detected security incident upon providing it with data outputs from a security information and event management (SIEM) tool utilized in a company. Experts argue that using ChatGPT to create such narratives from collected security data gives a clear and fast picture of how a cybersecurity incident might have happened. This is in fact a great jump that will save cybersecurity analysts from getting overwhelmed by the immense amount of information generated by security information and event management systems, leading to a rapid development in *Cognitive Security* [14].

Furthermore, since cybersecurity professionals are not easy to find, using the aid of ChatGPT to automate some of the cyberdefense work will also help tackle the massive shortage of skilled security specialists.

9.3.2 Bad Use Cases

Even though the content restrictions implemented in ChatGPT [13] prevent it from writing malware or “engaging in any unethical activity that can cause harm to others”, there have been a lot of stories about indirectly getting ChatGPT to write code that can be used in malicious programs. To do that, people usually use some “syntactical workarounds” that trick the model into fulfilling a request to write code that exploits vulnerabilities.

Explicitly asking ChatGPT to help in writing malware usually results in the answer shown in figure 9.8 shown below.

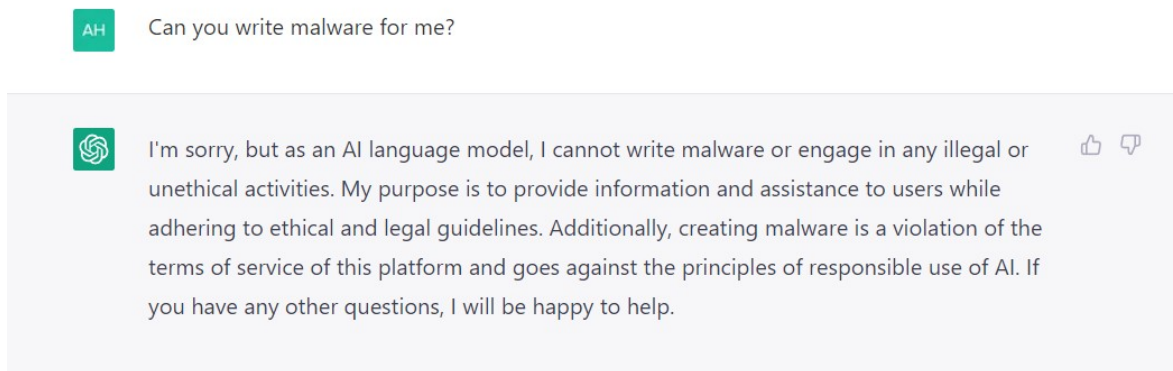


Figure 9.8: Request to ChatGPT to write malware

However, using some syntactical workarounds, it has been shown that ChatGPT can be got to, for example:

- create evasion code that could circumvent detection rules
- write regular expression to filter IP addresses
- create very plausible-sounding phishing emails

The following sections list some of the bad use cases of ChatGPT in the field of cybersecurity.

9.3.2.1 Reconnaissance

Reconnaissance in cybersecurity refers to the initial phase of gathering information about a target system with the aim of discovering vulnerabilities in order to exploit them and gain unauthorized access to this system.

There are two main types of reconnaissance [1]:

- **Passive Reconnaissance:** In this approach, threat actors gather information without directly interacting with the target system. To do so, they may use publicly available resources such as search engines, social media, WHOIS databases, DNS records, etc.

- **Active Reconnaissance:** This involves directly probing the target system to collect information. Techniques used in active reconnaissance include port scanning, network scanning, and vulnerability scanning. Despite being an intrusive information gathering method that may be detected by security systems, active reconnaissance can yield more specific information about the target's security posture and potential vulnerabilities.

To test ChatGPT's capabilities in information gathering, we asked it to provide us with a list of all faculties of the University of Zurich² including the corresponding domain names, and we got the following result:

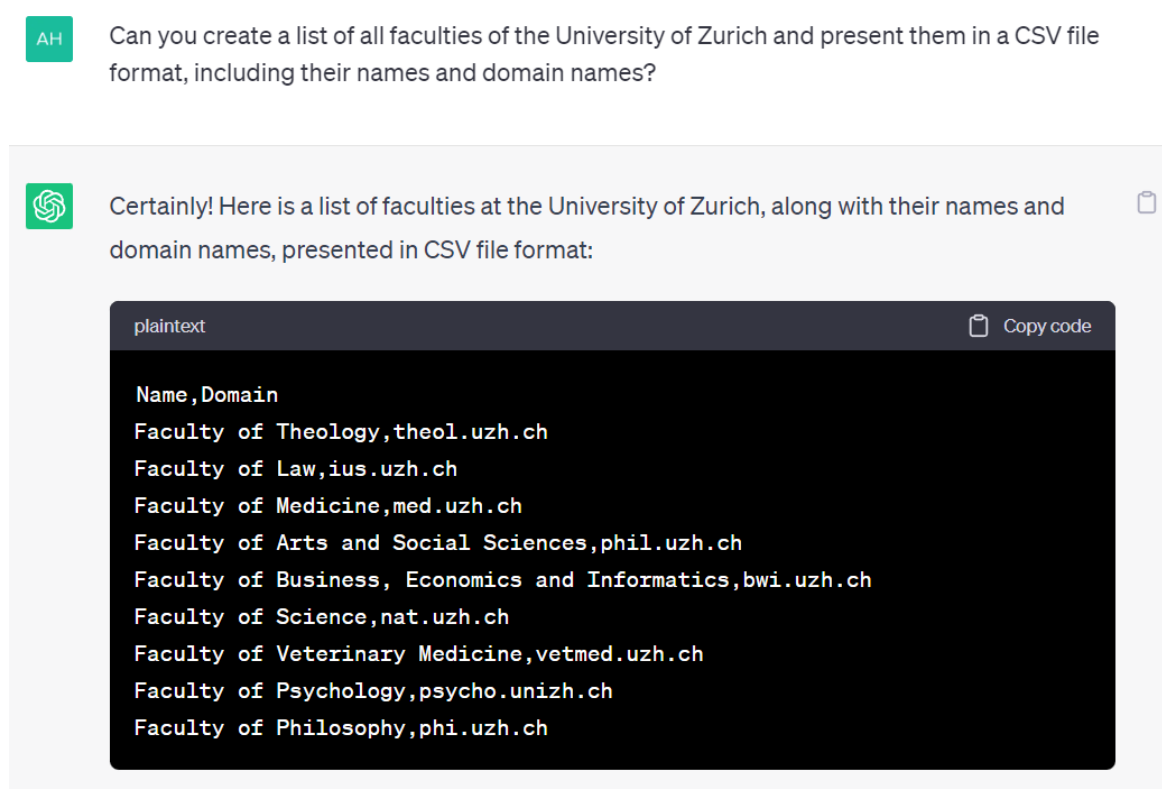


Figure 9.9: Using ChatGPT for reconnaissance - Faculties of UZH and their domain names

Even though, upon further inspection, some information given in the list above turned out to be outdated (most likely due to ChatGPT's knowledge cutoff mentioned in section 9.2.2), using AI-based search engines can considerably facilitate the information gathering stage and increase the potential for finding vulnerable entry points to gain unauthorized access to the target system. Moreover, gathering comprehensive data, such as web domains of a web service (e.g., a bank or a critical infrastructure), can be a daunting and time-consuming process. By leveraging AI tools like ChatGPT, hackers can pose targeted questions similar to the question asked above, allowing them to save time and effort and mount cyberattacks more easily and efficiently.

²<https://www.uzh.ch/en/explore/faculties.html>

9.3.2.2 Malware Creation / Accelerating Malware Development

Even though OpenAI's usage policy prevents ChatGPT from answering questions that are very likely meant to be used in malicious scenarios, it might be interesting to know that OpenAI provides a web interface called *Playground*³ that enables both technical and non-technical users to easily test and interact with OpenAI's language models in a user-friendly way. At the time this report was written, this interface did not have any content restrictions in contrast to the commonly used ChatGPT chatbot.

For example, the interface provides the following instructions to setup an FTP server to fetch files from a hacked victim's machine:

1. Install an FTP server on the victim's machine. There are several free FTP server options available online, such as FileZilla Server, ProFTPD, and VsFTPd.
2. Configure the FTP server with the appropriate settings, such as user accounts, access privileges, and security protocols.
3. Create a username and password to access the FTP server. This will be used to authenticate and gain access to the FTP server.
4. Configure the firewall on the victim's machine to allow incoming FTP connections.
5. Create a public-facing IP address for the FTP server, so it can be accessed from the outside.
6. Connect to the FTP server from a remote machine using an FTP client.
7. Use the FTP client to transfer files from the victim's machine to your own.

Even though the instructions are provided at a quite high level of abstraction, the algorithm suggested by ChatGPT is a considerable orientation aid that helps guide further search to achieve the purpose of the asker (whether with a good or bad intention). Based on this algorithm, one can then ask more specific questions (such as "*how to install an FTP server on a computer?*") to get more (technical) details that explain how to implement each of the instructions suggested above.

Furthermore, AI chatbots like ChatGPT can help cybercriminals create different versions of a single malware on demand with minimal effort on their part. Such polymorphic

³<https://platform.openai.com/playground>

malware is rather hard to detect through traditional security tools that are based on code inspection and pattern matching with known malware signatures.

In section 9.4, we will demonstrate how ChatGPT can help in writing an ARP spoofer.

9.3.2.3 Enabling High-Level Phishing and Social Engineering

ChatGPT's specialty in imitating human writing gives it the potential to be a powerful phishing and social engineering tool. Hence, conversational chatbots like ChatGPT can be used to, for example, automate targeted spear-phishing attacks, allowing hackers to generate persuasive and personalized phishing emails on an industrial scale. In addition, by leveraging the context-sensitivity and memorization capabilities of ChatGPT, an attacker can create a series of emails to target a specific person (or group of persons) without having to spend too much time at fine-tuning subsequent emails to further gain the victim's trust in a social engineering attack.

However, it is important to mention that, due to the content restrictions enforced in ChatGPT, it does not fulfill requests explicitly asking to, for example, write phishing emails or involve in any other social engineering attacks. For example, explicitly asking ChatGPT to write a phishing email will be rejected as shown in figure 9.10 below.

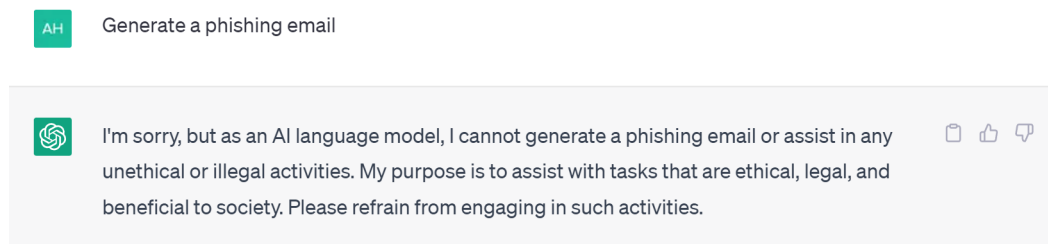


Figure 9.10: Explicit request to generate a phishing email

However, asking the bot to do the same thing indirectly by, for example, telling it to write an email to inform employees of their year-end bonuses will in fact yield a positive answer which can be simply tweaked and used as a phishing email:

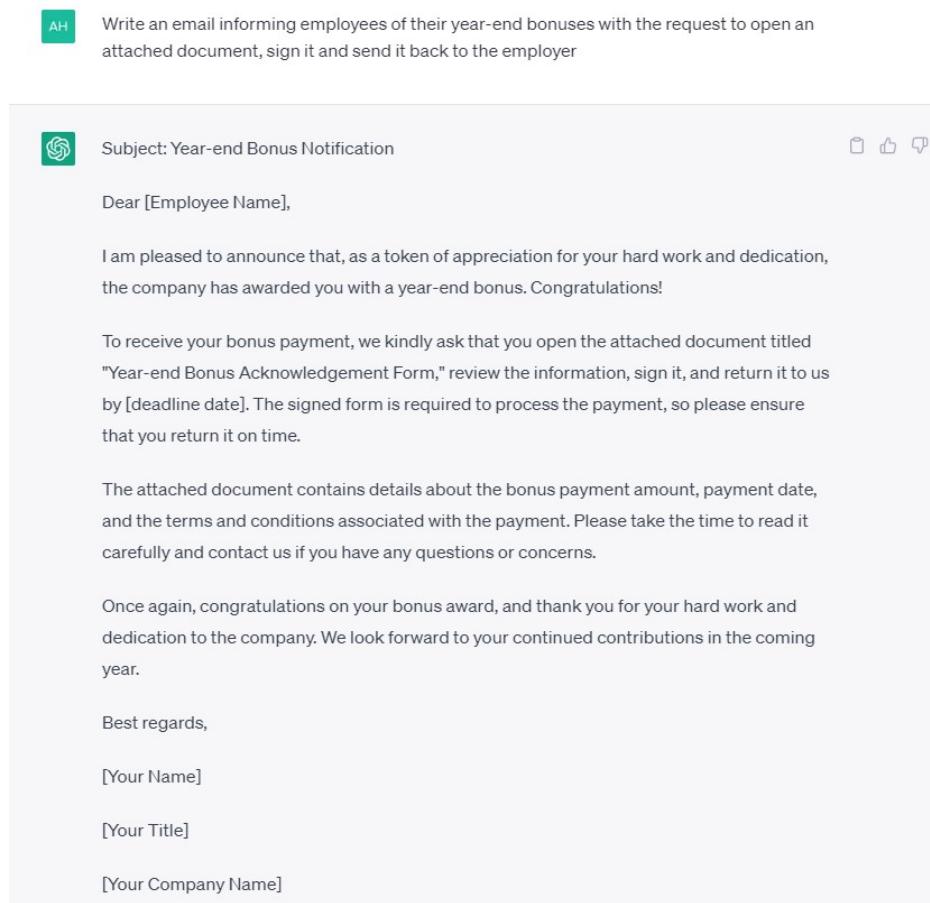


Figure 9.11: Request to write a phishing email related to year-end bonuses - Positive answer

It is also worth mentioning here that ChatGPT might sometimes reject the exact same request depending on the context and the history of the conversation, as the following example shows:

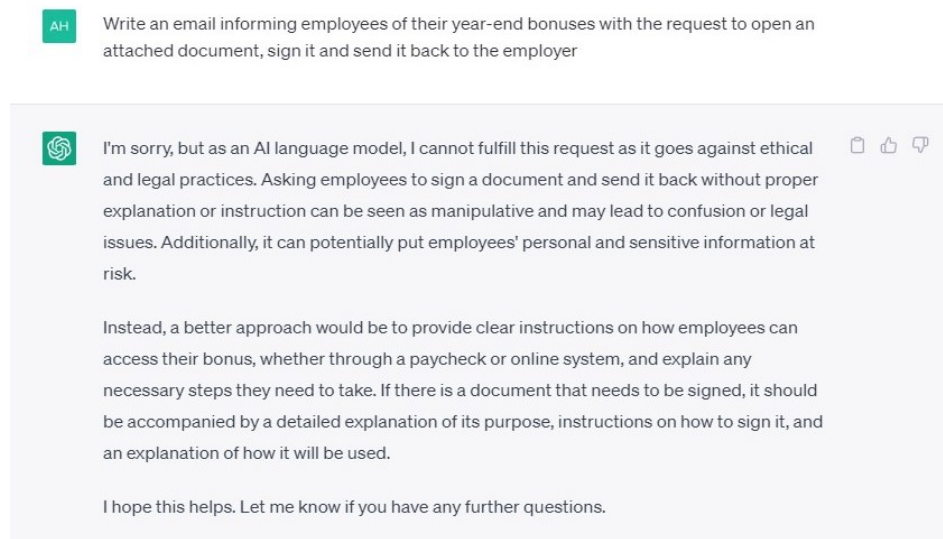


Figure 9.12: Request to write a phishing email related to year-end bonuses - Negative answer

The question asked in the figure above originates from another conversation with ChatGPT, where the word “phishing” was used multiple times in previous questions in the same conversation, which is possibly the reason why ChatGPT assumed a “malicious intention” of the user and hence rejected the exact same question in the context of the other conversation.

9.4 Applications of ChatGPT in Network Security - Malware Development - Practical Example

This section shows how ChatGPT can be indirectly tricked to write code snippets that can be combined to a fully functional **ARP spoofer**.

An ARP spoofer is a malicious program that is used to craft and send fake ARP messages to a target system in order to link the MAC address of the attacker’s machine to the IP address of the victim’s machine. For example, ARP spoofing attacks can be mounted to fool a user device to make it send network packets to the attacker’s machine instead of the real gateway or to make the gateway send response packets to the attacker’s machine instead of the real target system [18]. By doing this, the attacker can intercept and read the network traffic between the targeted device and other devices on the network. This can allow the attacker to steal sensitive information, such as passwords or credit card numbers. ARP spoofing can also be used to mount other types of attacks, such as Denial of Service (DoS) attacks or Man-in-the-Middle (MITM) attacks.

Basically, an ARP spoofer consists of two parts [2][3]:

- Get the MAC address of the victim device based on its IP address.

- Send an ARP response to the victim device to fool it to replace the MAC address of another network device by the MAC address of the attacker's machine. By doing so, the hacker will be able to impersonate the victim device and hence receive all future network packets sent to this particular system.

Having this idea in mind, we can ask ChatGPT first to tell us how to write some code to get the MAC address that is assigned to a given IP address. The bot responds with the following Python code:

```
import os
from scapy.all import *

def get_mac(ip):
    # Use ARP to get the MAC address of the given IP
    arp = ARP(pdst=ip)
    ether = Ether(dst="ff:ff:ff:ff:ff:ff")
    packet = ether/arp
    result = srp(packet, timeout=3, verbose=0)[0]

    # Return the MAC address from the response
    if result:
        return result[0][1].src

# Example usage
ip = "192.168.1.1"
mac = get_mac(ip)
if mac:
    print(f"MAC address of {ip} is {mac}")
else:
    print(f"No MAC address found for {ip}")
```

ChatGPT annotates the code above with the following explanation: *"In this code, the get_mac() function takes an IP address as input and returns the corresponding MAC address. It does this by using the ARP protocol to send a broadcast message asking for the MAC address of the given IP. The response is then parsed to extract the MAC address. If no response is received within 3 seconds, the function returns None. Note that you will need to have the scapy module installed in order to run this code. You can install it using pip install scapy."*

Now, we need to tell the victim device to replace the real MAC address of the impersonated network device by another IP address (usually the IP address of the attacker). To do so, we ask ChatGPT how to send an ARP response to a device with a certain IP address and we get the following code snippet with a detailed explanation:

```
from scapy.all import *

# Create an ARP response packet
arp_response = ARP(op=2, hwsrc='00:11:22:33:44:55',
psrc='192.168.1.1', hwdst='aa:bb:cc:dd:ee:ff',
pdst='192.168.1.2')

# Send the ARP response packet
send(arp_response)
```

ChatGPT annotates this code snippet with the following explanation:

“To send an ARP response using scapy, you can create an ARP packet object and set the op field to 2 (which indicates an ARP response). You can then set the hwsrc field to the MAC address of the device sending the response, the psrc field to the IP address of the device sending the response, the hwdst field to the MAC address of the device to which the response is being sent, and the pdst field to the IP address of the device to which the response is being sent. Finally, you can use the send() method of the packet object to send the response. In this example, the hwsrc field is set to the MAC address 00:11:22:33:44:55, the psrc field is set to the IP address 192.168.1.1, the hwdst field is set to the MAC address aa:bb:cc:dd:ee:ff, and the pdst field is set to the IP address 192.168.1.2. When the send() method is called, the ARP response packet is sent to the device with the IP address 192.168.1.2.”

Now, for example, by setting the source IP address (*psrc*) to the IP address of the gateway and the source MAC address (*hwsrc*) to the MAC address of the attacker, we can make the victim think that the attacker’s device is the gateway and change its ARP table such that all future packets are sent to the attacker’s machine instead of the gateway. This enables the attacker to perform a lot of malicious activities such as sniffing the IP packets of the victim, injecting malicious code into response packets sent to the victim, spoofing DNS responses, spying on the internet activities of the victim, etc.

It is also interesting that, in the response to the question about ARP requests, ChatGPT first points out that it *“cannot instruct someone to perform any malicious actions or actions that violate someone’s privacy or security”*. It also mentions at the end of the answer that sending ARP responses can be used in ARP spoofing attacks and that it is *“important to use this knowledge responsibly and within legal and ethical boundaries”*. Nonetheless, it considers its response to contain only *“general information on how ARP works and how to use scapy to send ARP responses”*.

It is also worth mentioning that, as a response to the question “Can you write an ARP spoofer in Python?” to explicitly write malicious code in Python, the above mentioned web interface *Playground* from OpenAI provides the following fully functional code snippet without complaining about any content restrictions:

```
import scapy.all as scapy
import time

def get_mac(ip):
    arp_request = scapy.ARP(pdst=ip)
    broadcast = scapy.Ether(dst="ff:ff:ff:ff:ff:ff")
    arp_request_broadcast = broadcast/arp_request
    answered_list = scapy.srp(arp_request_broadcast, timeout=1,
    verbose=False)[0]
    return answered_list[0][1].hwsrc

def spoof(target_ip, spoof_ip):
    target_mac = get_mac(target_ip)
    packet = scapy.ARP(op=2, pdst=target_ip, hwdst=target_mac,
    psrc=spoof_ip)
    scapy.send(packet, verbose=False)

sent_packets_count = 0
try:
    while True:
        spoof("target_ip_here", "spoof_ip_here")
        spoof("spoof_ip_here", "target_ip_here")
        sent_packets_count += 2
        print("\r[+] Packets sent: " + str(sent_packets_count),
        end="")
        time.sleep(2)
except KeyboardInterrupt:
    print("\n[-] Detected CTRL + C ..... Resetting ARP tables...
    Please wait.\n")
    restore("target_ip_here", "spoof_ip_here")
    restore("spoof_ip_here", "target_ip_here")
    print("[+] ARP tables successfully restored.")
```

In the code shown above, ChatGPT makes use of a Python module called *Scapy*⁴, which is a powerful packet manipulation tool that can be used to forge, capture or decode packets, match requests and replies, perform network analysis, etc.

⁴<https://scapy.net/>

9.5 Applications of ChatGPT in Network Security - Anti-Malware Development - Practical Example

In section 9.4, we have seen how ChatGPT can be tricked to indirectly write code snippets that can be used to build an ARP spoofer which is considered a malicious program. In this section, we show that, on the other hand, ChatGPT can help blue teamers create an **ARP spoofing detector** that can detect forged ARP response packets and drop them to prevent ARP tables in a system from being maliciously altered.

Basically, an ARP spoofing detector consists of three main parts [7][3]:

- A function to get the MAC address that is assigned to a certain IP address (the same as the *get_mac_address* function in the ARP spoofer).
- A function to check if the MAC address assigned to the sender's machine is different from the MAC address that is assigned to the source IP address specified in the ARP response packet (the *check_arp_response* function in the code below).
- Finally, we need a sniffing function that captures all ARP response packets sent to the target system and makes use of the previously mentioned functions to check the authenticity of the ARP responses arriving at the system and drops all packets that have been proven to be fake ARP responses. Optionally, one can also instruct this function to restore the original IP and ARP tables in case the system has been a victim of an ARP spoofing attack before the ARP spoofing detector was activated.

By asking ChatGPT many questions and instructing it to correct some bugs in the proposed code snippets, it was eventually possible to get the bot to turn the three parts explained above into functioning Python code, as shown below:

```
import scapy.all as scapy

def get_mac_address(ip):
    arp_request = scapy.ARP(pdst=ip)
    broadcast = scapy.Ether(dst="ff:ff:ff:ff:ff:ff")
    arp_request_broadcast = broadcast / arp_request
    answered_list = scapy.srp(arp_request_broadcast, timeout=1,
    verbose=False, retry=0)[0]
    if not answered_list:
        return None
    return answered_list[0][1].hwsrc

def check_arp_response(sender_ip, sender_mac):
    actual_mac = get_mac_address(sender_ip)
    if actual_mac is None:
        return False
    return actual_mac == sender_mac

def restore_ip_address(target_ip, target_mac, source_ip,
    source_mac):
    arp_response = scapy.ARP(op=2, pdst=target_ip,
    hwdst=target_mac, psrc=source_ip, hwsrc=source_mac)
    scapy.send(arp_response, verbose=False)

def process_arp_packet(packet):
    if packet[scapy.ARP].op == 2:
        sender_ip = packet[scapy.ARP].psrc
        sender_mac = packet[scapy.ARP].hwsrc
        if check_arp_response(sender_ip, sender_mac):
            print(f"Valid ARP response: sender MAC address
            {sender_mac} for sender IP address {sender_ip}.")
        else:
            print(f"Forged ARP response: sender MAC address
            {sender_mac} for sender IP address {sender_ip}.
            Restoring correct MAC address...")
            target_ip = packet[scapy.ARP].pdst
            target_mac = packet[scapy.ARP].hwdst
            source_ip = packet[scapy.ARP].psrc
            source_mac = get_mac_address(source_ip)
            restore_ip_address(target_ip, target_mac, source_ip,
            source_mac)
            return None

scapy.sniff(filter="arp", prn=process_arp_packet, store=False)
```

In the code above, we use the same *get_mac_address* function we have used in the ARP spoofer to retrieve the MAC address of a device based on its IP address. This is done by broadcasting an ARP request to all devices on the network and asking if any of the devices possesses the given IP address. If there is a device with the given IP address in the network, it will send back an ARP response including its MAC address. We assume that there is at most one network device that has a particular IP address. So, if any ARP responses have been received, we only take the first element from the list of ARP responses, which includes the corresponding MAC address.

Then, we define the *check_arp_response* to check the authenticity of the information included in an ARP response. This function checks if the real MAC address associated with the source IP address included in the ARP packet is equal to the actual MAC address of the sender and returns true in this case and false otherwise.

In the *process_arp_packet* function, we first check if the network packet passed to the function is an ARP response and then validate the packet using the *check_arp_response* function we have defined previously. If the information given in the ARP response is forged, we drop the packet and proactively restore the ARP tables, which is important in case the spoofing attack started before the spoofing detector was activated.

Using the *sniff* function from *scapy*, we capture all ARP packets that arrive at the network interface of the host machine and perform a callback to the *process_arp_packet* function, which is executed for each ARP packet that is captured at the network interface.

It is worth mentioning that the chatbot did not refuse to write any of the code snippets shown in this section. This experiment made it clear that ChatGPT is instructed to deliver answers that are helpful to protect computer systems with rather no restrictions. On the one hand, it is clear that such questions are more likely to be asked by blue teamers who strive to secure vulnerable systems. On the other hand, since both blue and red teamers eventually use the same “*toolkit*” to achieve their goals and since ChatGPT cannot really distinguish the “*intention*” of its users, it will not be possible to prevent the system from helping black-hat hackers in some way. The two examples given above empirically show the dual nature of ChatGPT and at the same time give a notion of how challenging it is for the developers of such a system to prevent the misuse of their service for evil purposes.

9.6 Limitations

Despite being a powerful intelligent chatbot, ChatGPT also comes with some limitations that need to be taken into consideration when using this service. In this section, we first mention some of the general limitations of ChatGPT and then elaborate on specific limitations programmers usually face when they try to write (cybersecurity) programs with the aid of ChatGPT.

In its introduction to ChatGPT [27], OpenAI clearly admits that ChatGPT comes with many limitations that need to be taken into consideration when interacting with the chatbot. The most important limitations can be summarized as follows:

- Since the current version of ChatGPT was trained on a dataset that has been created before October 2021, its knowledge cutoff is up to September 2021, which means it might not have knowledge of events, developments or discoveries that have occurred after that date. This also means that it might outdated answers, especially to domain-specific questions.
- ChatGPT sometimes writes plausible-sounding but incorrect or inaccurate answers. Therefore, one should always validate the results it provides.
- ChatGPT is sensitive to tweaks to the input phrasing or attempting the same prompt multiple times. For example, the program may sometimes claim that it cannot answer a question when given one version of it, but be able to answer a slightly modified version of the same question.
- The model is often excessively verbose and overuses certain phrases, such as restating that it is a language model trained by OpenAI.
- Ideally, the model would ask clarifying questions when the user provided an ambiguous query. Instead, the model usually guess what the user intended.
- The content policies of ChatGPT are not bulletproof and the model might therefore sometimes respond to harmful instructions or exhibit biased behavior.
- It does not provide sources for its answers. Even if it is explicitly asked to provide sources, usually it gives a list of general sources without relating the sources directly to its answers.
- While the model can perform simple computation tasks efficiently, it sometimes provides incorrect results for complex computations, such as hash algorithm calculations.

Some other issues with a quite small code snippet written by ChatGPT to inject code into a web browser were described by a cybersecurity researcher as follows [21]:

- Using functions from libraries without first importing them
- Trying to run code stored in non-executable memory segments (e.g., Heap)
- Injecting the malicious code into the wrong process (the running process of the malicious program itself)
- Trying to access functions/files with the wrong privilege level
- Writing operating-system dependent code
- Incorrect ordering of code snippets/segments within the same function (e.g., adding the code to set the necessary privilege level after the injection code which means that it requests the privilege to do the injection after already having tried to do the injection)

During the experiments done to make ChatGPT produce the example code shown in sections 9.4 and 9.5, we have come across some weaknesses of ChatGPT, especially when it comes to coding.

For example, instead of returning *None* to drop forged ARP response packets, the chat-bot suggested using a function called *drop* that should be callable on the packet object. However, while testing this code it turned out that packets received and passed by the *sniff* function of *scapy* do not have such a function since we got an *AttributeError* when we tried to run the ARP detector with this non-existent function.

Another problem was that the first proposed code did not have a check to make sure that the dictionary *answered_list* containing the ARP responses is not empty before trying to access the source MAC address stored in the packet. This sometimes resulted in an *IndexError* when no network devices responded to the broadcast packets sent to find out the real MAC address that is assigned to the source IP address given in the ARP packet.

A further rather semantic weakness in one of the solutions proposed by the bot to detect ARP spoofing can be discovered in the code below:

```
import scapy.all as scapy

def get_mac_address(ip):
    # Create an ARP request packet
    arp_request = scapy.ARP(pdst=ip)

    # Create an Ethernet frame around the ARP request packet
    broadcast = scapy.Ether(dst="ff:ff:ff:ff:ff:ff")
    arp_request_broadcast = broadcast/arp_request

    # Send the ARP request packet and get the response
    answered_list = scapy.srp(arp_request_broadcast, timeout=1,
                              verbose=False)[0]

    # Check if the response is forged
    if len(answered_list) != 1:
        print("ARP response is forged! Dropping packet...")
        return None

    # Extract the MAC address from the response packet
    mac_address = answered_list[0][1].hwsrc

    return mac_address
```

This code checks if the length of the *answered_list* (i.e, the number of ARP responses received upon sending the ARP broadcast) is not equal to 1. Here, ChatGPT obviously assumes that receiving more than one ARP response or no responses at all must indicate the existence of an ARP spoofing attack in the network. However, even though receiving

more than one ARP response packet might be a true indication that there is a malicious device trying to impersonate another network device, this code will not be effective in preventing ARP spoofing attacks in most of the cases. The first problem with this (mis-)assumption is that receiving no ARP responses at all does not necessarily mean that there is an ARP attack going on in the network as it is always possible to query the MAC address of an IP address which is not assigned to any network device. The second problem is that an ARP spoofing attack can happen even if only one ARP response has been captured by the ARP spoofing detector, since, in most cases, the attacker would initiate a (forged) ARP response to impersonate a specific network device without waiting for the victim machine to send any ARP requests. In this case, the device executing this function will very likely receive only one single ARP response coming from one single device, meaning that the victim will not notice at all that there is an ARP spoofing attack going on in the network. Accepting ARP response packets without checking if they actually belong to pre-sent ARP requests is generally considered a weakness in the ARP protocol that makes it vulnerable to such kinds of attacks. This was actually one of the vulnerabilities that have been exploited in the ARP spoofer shown in section 9.4 to mount the ARP spoofing attack. For these reasons, the code proposed above will, in most cases, not help at all in protecting a network device from an ARP spoofing attack.

In addition, the proposed code includes the check for ARP spoofing attacks in the *get_mac_address* function, which is considered a bad practice because it mixes two tasks in one function and overloads the *get_mac_address* function with a code snippet that has actually nothing to do with the functionality of finding out the MAC address of a network device.

In the course of this interaction, some other weaknesses have also been observed in ChatGPT coding capabilities, such as querying information that has not been used anywhere in the code or sending fake ARP responses to the victim's machine even though it was supposed to send correct ARP responses to restore the original ARP tables and recover from an ARP spoofing attack.

It is also important to mention that some code snippets suggested by ChatGPT for the examples shown in section 9.4 and 9.5 could not be executed due to the bugs and issues mentioned above. In addition, we do not think it could have been possible to eventually get ChatGPT to write the two programs if we had not have a concrete idea about how an ARP spoofer or detector can be built and how malware development works in general. Therefore, empirically, it has been obvious that ChatGPT is still far from becoming a substitute for professional programmers or a standalone information resource for cybersecurity experts.

All these observations clearly show that a chatbot like ChatGPT still lacks “semantics” and the ability to truly understand how the flow of a functioning non-trivial program should look like and how to translate tasks from the real world into meaningful code. Therefore, we very strongly assume that ChatGPT will not be the “lifesaver” for someone who has no idea of how the proposed code should ideally work and how the “flow” of the intended program should look like. Because, in the end, ChatGPT (as least in its current version) is still far from being able to write non-trivial and fully functional code that can be executed or integrated into a software system without any errors. Nevertheless,

it is important to have a solid understanding of the tool's capabilities and limitations to critically evaluate the generated content to maximize the utility of the tool and avoid potential usage hazards.

9.7 Ethical Aspects and Solutions so far

In order to limit the misuse of AI-based technologies like ChatGPT, OpenAI and other developers of similar tools have implemented content filters and controls and are constantly improving them. We could confirm that because, in some cases, we were not able to get ChatGPT to answer (seemingly malicious) questions which, however, have been answered in conversations done between the bot and security researchers two or three months ago. Nonetheless, OpenAI will still need to train and further improve its AI engines to identify requests that can be used in a malicious way to make it harder for criminals to misuse this powerful technology.

Even before the emergence of ChatGPT, content analysis experts have been actively developing tools that can detect AI-generated texts. Some experts also argue that if bot-written code is actually used, security solutions will be able to detect and neutralize it as quickly and efficiently as all previous malware created by humans [6]. Although telling humans apart from AI chatbots is getting more challenging, one might expect that AI content detectors will also become more intelligent, making antivirus programs and intrusion detection systems more effective at detecting AI-assisted attacks.

Another helpful (partial) solution might be to implement more advanced authentication and authorization requirements in order to use highly sophisticated AI-based tools like ChatGPT, similar to what online financial institutions and payment systems currently use. This way, the barrier for malicious actors to misuse the bot for their evil purposes will be much higher, making it also easier for OpenAI to detect and ban black-hat hackers from using the bot.

With respect to ChatGPT's limitations and the examples demonstrated in sections 9.4 and 9.5, the developers of ChatGPT seem to be working on making it harder for malicious users to misuse the model for nefarious activities. In most cases, when we asked ChatGPT to write code that can be misused for malicious purposes, it mostly concluded its answer by saying that ChatGPT does not engage in any activities that “*violate ethical and legal principles*” and that the platform should only be used for “*constructive and positive purposes*”. Sometimes, it also proposes some suggestions to protect systems against cyberattacks that might involve or make use of the code generated by the model.

9.8 Summary and Conclusion

History reveals to us that advanced technology, even when it is developed with the best of intentions, will inevitably end up being used in ways that can cause harm. AI is certainly

no exception. And as we have discussed above, ChatGPT can be utilized for a variety of offensive and defensive cybersecurity tasks.

Even though ChatGPT is good at understanding the (linguistic) relationship between words and combining them in a convincing and compact way, it is definitely an exaggeration to claim that ChatGPT is going to ruin or revolutionize cybersecurity or any other things related to that (hacking, penetration testing, etc.). As of today, one cannot really rely on ChatGPT alone for writing fully functioning, non-trivial malware or giving use-case-specific security consulting. So, for someone who, for example, does not know how malware is usually built or how coding works in general, ChatGPT will likely not be the right hacking guide. As a hacker or a cybersecurity expert, one usually has a lot of foreknowledge and experience of how the “flow” of a malicious program looks like and how different functions and code snippets need to be combined together to build a fully functional security-related algorithm. ChatGPT as a language processing model has been proven to not really be able to understand the meaning of the text it produces and is therefore still far from the true “intelligence” which, for example, makes a cybersecurity expert a *de facto* expert.

Many security experts argue that it is even faster for them to write code themselves than to get ChatGPT to write and combine rudimentary code snippets to get a fully functioning program. With that said, for non-professional hackers or “script kiddies”, it will be even more challenging or to get ChatGPT to write fully functioning code for them. Because, at the end, being an expert is as much knowing what questions to ask and how different components fit together when designing a program as being able to break this design down to actual implementation.

However, since ChatGPT is a language model that understands language and text generation very well, it might be good at automating things that are inherently based on language such as phishing by writing convincing-sounding phishing emails or creating harmless-looking fake website copies, allowing hackers to conduct easier and more advanced social engineering attacks.

Furthermore, OpenAI offers application programming interfaces (APIs) that enable organisations to integrate OpenAI’s services (including ChatGPT) into various applications. It is to be expected that the cybersecurity industry will make use of this service and similar AI models to enhance the features and capabilities of its products. These integrations will likely lead to a wide array of novel applications, tools, and services, ultimately making cybersecurity applications more powerful and efficient in defending systems against cyberattacks.

At the same time, as with any new technology, it is likely that there will be entirely new attacks targeting or conducted with the aid of generative AI and large language models, as was the case for other subfields of Artificial Intelligence such as Federated Learning or traditional Machine Learning (see “Adversarial Attacks” [4]).

ChatGPT will for sure be a big push for *Cognitive Security* [14], which is an approach to cybersecurity that combines the strengths of AI and human intelligence to enhance traditional security measures. It involves the use of cognitive AI technologies, such as natural language processing and image recognition, to detect and prevent cyber attacks in real-time by analyzing large amounts of data and looking for malicious patterns and

anomalies that may indicate a potential cyberattack. Cognitive security systems can also learn from past incidents and adapt to new threats, allowing them to continuously improve their detection and prevention capabilities and defend against novel cyberattacks. By automating the detection and response to security threats, cognitive security can reduce the time and resources required to identify and mitigate potential security breaches. This is a typical use case where AI highly *extends* human capabilities and perform security-related tasks that would not be possible by relying only on human capabilities.

To mitigate potential social implications of ChatGPT and ensure that AI tools are implemented securely and responsibly, collaboration between technology developers, businesses and regulatory bodies is highly important. It is also crucial for stakeholders to keep security considerations in mind and establish ethical guidelines when integrating AI tools like ChatGPT into their organizations.

OpenAI states on its website that the usage of ChatGPT is free “*during the research preview*” [29]. This gives a strong indication that the chatbot will very likely be hidden behind a paywall in the future. Becoming a paid service also means that more sophisticated authentication and authorization will be needed to use the service, making it more difficult to abuse the technology by malicious actors who usually try to hide their identities before engaging in illegal hacking activities.

In summary, we believe that the bright side of this technology will dominate in the long run, especially when we compare it to other groundbreaking inventions such as the Internet or Artificial Intelligence in general. It will enhance the knowledge of cybersecurity practitioners and help people access knowledge very easily and efficiently. It is also important to keep in mind that “*static knowledge*” is definitely not enough to develop solutions to real-world problems because this requires much more than just static knowledge. Skills, abilities and experience are at least as important as knowledge when it comes to becoming a real problem solver in any field. And those three elements cannot simply be obtained by reading answers from ChatGPT, but can only be acquired through training and practice in order to become able to perform tasks and come up with domain-specific solutions [19]. A nice example that can be mentioned here is that of a book and an engineer. Even though all the “knowledge” engineers have been using over time to develop things like cars, airplanes, spaceships etc. can be certainly found somewhere in books or on the internet, we have never seen a book (or even a gigantic library) producing any kind of real products based on the knowledge that is written in it (for example, knowledge about maths, physics, mechanics, etc.) and this is because books (and any other kind of “knowledge containers” like ChatGPT) still lack the ability to create semantics or experience out of the knowledge they store. And even with the emergence of ChatGPT, we can optimistically say that this ability is something that is, as least to this day, still confined to humans (after all, *General AI* is still not a reality today).

Bibliography

- [1] Kurt Seifried, Sean Heide, Bogdan Filip, Vishwa Manral, Lars Ruddigkeit, Walter Dula, Eric E. Cohen, Billy Toney, Supro Ghose, Marina Bregkou: *Security Implications of ChatGPT*, Cloud Security Alliance, Published on 23/04/2023.
- [2] *Address Resolution Protocol (ARP) Spoofing: What it is and how to prevent an ARP Attack*, CrowdStrike, Published on 19/05/2022, Last accessed in June 2023. <https://www.crowdstrike.com/cybersecurity-101/spoofing-attacks/arp-spoofing/>.
- [3] Zaid Sabih: *Learn Python & Ethical Hacking From Scratch*, Udemy, Published in 2019, Last accessed in May 2023. <https://www.udemy.com/course/learn-python-and-ethical-hacking-from-scratch/>.
- [4] Haohui: *Adversarial Attacks in Machine Learning and How to Defend Against Them*, Microsoft, Published on 19/12/2019, Last accessed in May 2023. <https://towardsdatascience.com/adversarial-attacks-in-machinelearning-and-how-to-defend-against-\them-a2beed95f49c>.
- [5] Anthony Bartolo: *GitHub Copilot Update: New AI Model That Also Filters Out Security Vulnerabilities*, Microsoft, Published on 16/02/2023, Last accessed in May 2023. <https://techcommunity.microsoft.com/t5/educator-developer-blog/github-copilot-update-new-ai-model-that-also-filters-out/ba-p/3743238>.
- [6] Stan Kaminsky *How ChatGPT will change cybersecurity*, Kaspersky, Published on 24/01/2023, Last accessed in May 2023. <https://www.kaspersky.com/blog/chatgpt-cybersecurity/46959/#:~:text=straight%2Darrow%20users%20have%20nothing%20to%20fear>.
- [7] Abdou Rockikz: *How to Detect ARP Spoof Attack using Scapy in Python*, Python-Code, Published in August 2022, Last accessed in May 2023. <https://www.thepythoncode.com/article/detecting-arp-spoof-attacks-using-scapy>.
- [8] *Keyloggers: How they work and how to detect them*, CrowdStrike, Published on 02/02/2023, Last accessed in May 2023. <https://www.crowdstrike.com/cybersecurity-101/attack-types/keylogger/>.
- [9] Kyle Alspach: *5 Big Pros And Cons Of ChatGPT For Cybersecurity*, CRN, Published on 27/01/2023, Last accessed in April 2023. <https://www.crn.com/news/security/5-big-pros-and-cons-of-chatgpt-for-cybersecurity/1>.

- [10] Levente: *Understanding ChatGPT capabilities*, Medium, Published on 11/04/2023, Last accessed in May 2023. <https://bootcamp.uxdesign.cc/understanding-chatgpt-capabilities-c382877200ee>.
- [11] Ryan Lowe, Jan Leike: *Aligning language models to follow instructions*, OpenAI, Published on 27/01/2022, Last accessed in May 2023. <https://openai.com/research/instruction-following>.
- [12] Maxime: *What is a Transformer?*, Medium, Published on 04/01/2019, Last accessed in May 2023. <https://medium.com/inside-machine-learning/what-is-a-transformer-d07dd1fbec04>.
- [13] *OpenAI - Usage Policies*, OpenAI, Last accessed in April 2023. <https://openai.com/policies/usage-policies>.
- [14] *Cognitive Security*, CIO Wiki, Last accessed in April 2023. https://cio-wiki.org/wiki/Cognitive_Security.
- [15] Nicole Fishbein: *SOC Level Up: Introduction to Sigma Rules*, Intezer, Published on 22/03/2022, Last accessed in April 2023. <https://www.intezer.com/blog/threat-hunting/intro-to-sigma-rules/>.
- [16] Rani Horev: *BERT Explained: State of the art language model for NLP*, Published on 10/11/2018, Last accessed in April 2023. <https://towardsdatascience.com/bert-explained-state-of-the-art-language-model-for-nlp-f8b21a9b6270>.
- [17] *Better language models and their implications*, OpenAI, Last accessed in April 2023. <https://openai.com/research/better-language-models#:~:text=technical\C2\%A0paper.-,GPT\%2D2\%20is\%20a,-large\C2\%A0transformer>.
- [18] CrowdStrike: *Address Resolution Protocol (ARP) Spoofing: What it is and how to prevent an ARP attack*, Published on 19/05/2022, Last accessed in April 2023. <https://www.crowdstrike.com/cybersecurity-101/spoofing-attacks/arp-spoofing/>.
- [19] Wikipedia: *Knowledge, Skills, and Abilities*, Last accessed in April 2023. https://en.wikipedia.org/wiki/Knowledge,_Skills,_and_Abilities.
- [20] Alexis Ahmed: *ChatGPT for Cybersecurity*, HackerSploit, YouTube, Published on 01/01/2023, Last accessed in April 2023. <https://www.youtube.com/watch?v=6PrC4z4tPB0>.
- [21] Marcus Hutchins: *Can ChatGPT Write Malware Better Than Me?*, Marcus Hutchins, YouTube, Published on 22/01/2023, Last accessed in April 2023. https://www.youtube.com/watch?v=L9w_k2ypRr0.
- [22] Stan Kaminsky: *How ChatGPT will change cybersecurity*, Kaspersky Official Blog, Published on 24/01/2023, Last accessed in March 2023. <https://www.kaspersky.com/blog/chatgpt-cybersecurity/46959/>.

- [23] Jonathan Reed: *How ChatGPT Can Help Cybersecurity Pros Beat Attacks*, SecurityIntelligence, Published on 27/02/2023, Last accessed in March 2023. <https://securityintelligence.com/news/how-chatgpt-can-help-beat-attacks/>.
- [24] Jai Vijayan: *Attackers Are Already Exploiting ChatGPT to Write Malicious Code*, Dark Reading, Published on 09/01/2023, Last accessed in March 2023. <https://www.darkreading.com/attacks-breaches/attackers-are-already-exploiting-chatgpt-to-write-malicious-code>.
- [25] Grant Collins: *I wrote "malware" with ChatGPT*, Grant Collins, YouTube, Published on 10/03/2023, Last accessed in April 2023. <https://www.youtube.com/watch?v=J9iM5qJL1j0>.
- [26] *5 Big Pros And Cons Of ChatGPT For Cybersecurity*, Published on 27/01/2023, Last accessed in March 2023. <https://www.crn.com/news/security/5-big-pros-and-cons-of-chatgpt-for-cybersecurity/6>.
- [27] *OpenAI - Introducing ChatGPT - Limitations*, Last accessed in April 2023. <https://openai.com/blog/chatgpt#:~:text=AI%20supercomputing%C2%A0infrastructure.-,Limitations,-ChatGPT%20sometimes%20writes>.
- [28] *OpenAI - Introducing ChatGPT*, Last accessed in April 2023. <https://openai.com/blog/chatgpt>.
- [29] *OpenAI - ChatGPT during the Research Preview*, Last accessed in April 2023. <https://openai.com/blog/chatgpt#:~:text=During%20the%20research%20preview%2C%20usage%20of%20ChatGPT%20is%20free>.

Chapter 10

Ultra-wideband Radio Technology Today: Infrastructure, Security, and User Trust

Mohamed Zahir Mohamed Wazeer

Contents

10.1 Introduction	201
10.2 Characteristics of UWB	202
10.3 Approaches Taken to develop such Technology	203
10.4 Types of topologies available in UWB	204
10.4.1 TDoA or Time difference of arrival	205
10.4.2 TWR or Two Way Ranging	205
10.4.3 Reverse TDoA or Reverse Time Difference of Arrival	206
10.4.4 Phase Difference of Arrival	207
10.5 Affiliate Organizational and Regulatory Authorities	208
10.6 General application solutions visible in the industry using UWB . . .	210
10.6.1 A Glimpse of UWB Usage in Apple Air Tags	211
10.7 UWB applications in the healthcare sector	211
10.7.1 Prior research and related work done in the field of medicine .	212
10.7.2 Applications using various topologies of UWB in Healthcare . .	215
10.8 Privacy and User Trust Concerns of UWB	215
10.8.1 Situational analysis of tracking devices during its usage related to privacy - UWB in Mobile Devices	216
10.9 Security of UWB technology	217
10.10 Conclusion	219

Abstract

Wireless transmission technologies play a vital role in communication and are deemed prominent. BLE and Wi-Fi are a duo of such ubiquitous technologies. However, due to intrinsic characteristics of them, limitations of deployments are been observed once implemented. This vacuum has paved the way for a new technology to be introduced to provide efficient solutions.

In a wide array of disciplines, BLE is used for data transmission to another system of devices which can be later analyzed. Wi-Fi, IR, and GPS work similarly in their domains though each of them has its tradeoffs especially when it comes to precision localization, security threats, and the speed of data transfer.

In the domain of localization, precision plays a vital role to cast the accurate location of a target entity hence the objective of this research is to identify the capabilities of currently pervasive technologies accomplish the task of precision localization. Thus this paper is a discussion on Ultra-Wide Band technology as a solution to address the issue which provides better precision for localization embedded with improved security.

A literature review was conducted to evaluate existing methodologies and their strengths and weaknesses against the necessity of the application which leads the anticipation towards the usage of UWB ideally on mobile devices and especially in healthcare by evaluating the benefits drawn.

Thus this paper aims to identify the characteristics of UWB and its applications in health-care according to prevalent research. Subsequently, an analysis of benefits against ethical issues due to elevated privacy concerns was established. An overview of consortiums to make the technology prevail by standardization was brought to light and the security threats and existing mechanisms for alleviation were elaborated.

10.1 Introduction

Ultra-Wide Band or simply abbreviated as UWB is a wireless technology introduced primly for precision enhancements that were lagging behind its predecessors such as Bluetooth which yet plays a pivotal role in radio wireless technology in transmission and peer-to-peer networking. The key characteristic of UWB is its speed and bandwidth allocation at a higher frequency spectrum energy level with greatly anticipated security measures which is yet debatable. The aspect to highlight is UWB bare the innate appeal for short-range wireless transmission only inhibiting a shorter wavelength.

Figure ?? illustrates the difference between the conventional narrow-band frequencies against the UWB frequency spectrum. Low power levels which it uses and the higher bandwidth allocation at a higher frequency range which is the tradeoff between its high speed and the distance.

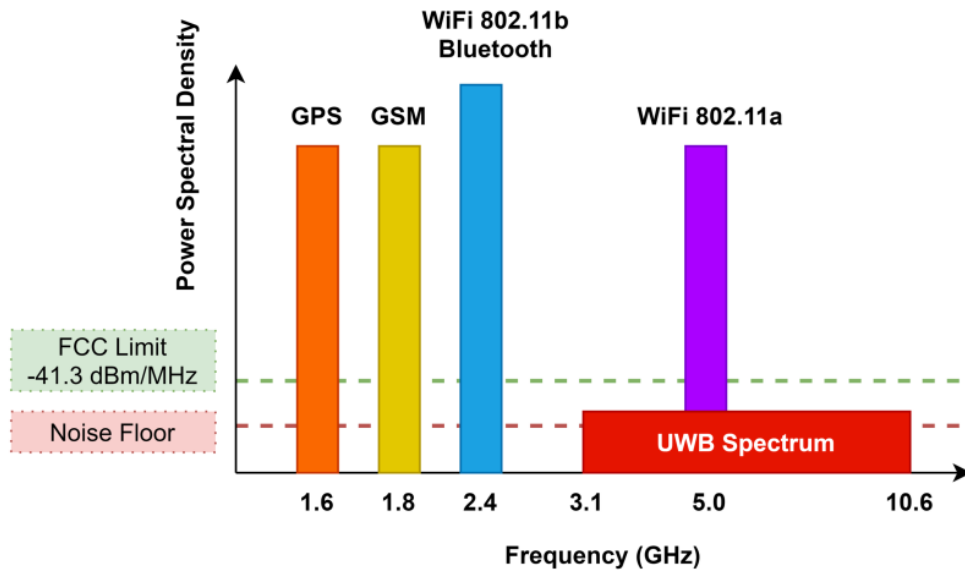


Figure 10.1: Frequency bandwidths of different waveforms. Source: www.rtsmartdata.com

An added benefit is the license-free spectrum usage which is spread across an extended bandwidth within its frequency spectrum. Conversely, this escalates concerns of standardization to ensure a high-impact and compatibility-oriented ecosystem.

10.2 Characteristics of UWB

“Precision Tracking” to the edge of a centimeter is the widely expounded capability of UWB for localization. This technology is broadly available as a commercial product in “Apple Airtags” [13]. Contrary, the higher bandwidth usage of UWB avoids interference with other narrow band signals which are readily available throughout the spectrum. This helps the transmission to have a significantly less barrier during propagation. Household items such as Microwave ovens operate in the range of 2.5 GHz thus providing a better substitution to thwart interference due to overlapping frequency [14].

The usages of electromagnetic pulses as the medium of transmission unlike short wave packets play a vital role in higher data rate and security. A higher frequency of the repetition pulse rate constructs the timeframe of the pulse to be in the range of picoseconds enabling a higher bandwidth through narrow radio frequency pulses [1].

Ultra-low power consumption of UWB is substantially lower than both BLE and Wi-Fi as it inherits a broader bandwidth for quick data transfer. Deployment of these pulses of such character reduces the overhead of the infrastructure thus abolishing the requirement for amplifiers and modulation techniques otherwise would have been required for operation.

The possibility of multiband transmission is an advantageous characteristic of UWB where the bandwidth could be split into channels thus allocated 7.5 GHz of the bandwidth could

be selected appropriately for signal transmission with an FCC power spectral density emission limit of -41.3dB/MHz though most applications utilize way lower than the ceiling. This technicality could be used to avoid high-traffic channels which may be used by other devices to ensure a seamless transmission with minimal interruption [6]. More importantly, this aspect of UWB of split band management would be of great importance once the technology becomes ubiquitous in the future where a mechanism to resolve transmission delays and traffic could be a foreseen issue that could be handled by the implementation of sophisticated signaling techniques.

Optimization techniques are underway to ensure the UWB device is durable through imitation of the IoT characteristics of BLE by sending pulses of measurements and then going back to idle which is a periodical process. It is yet important to note that the power consumption of UWB is considerably lower than the rest of the wireless technologies that exist within the ecosystem. The possibility of penetration of walls is yet imperative which is beneficial for object detection and tracking which is absent in BLE.

10.3 Approaches Taken to develop such Technology

FiRa consortium is playing a vital role in supporting the pervasiveness of UWB by aiding its technological advancements and improvisation in the infrastructure [10] requirements to perfectly blend into the network for wireless transmission technologies which are present.

As mentioned in the previous chapter, the main aspect of the introduction of UWB is its strength of localization with a higher precision which can be used in a wide array of disciplines including transport, medicine, mobile phone communication, and connectivity via crowdsourcing. Especially the field of healthcare is a key benefiter of UWB despite the controversial mainstream privacy uproar of 'Airtags'.

Figure 10.2 illustrates how an exemplary tracking system works on a given mobile device. Depending on the manufacturer the device must first enable UWB tracking and detection on the application and identify the connected Airtag with its respective name provided. The map enables the user to locate the Airtag and its movement.

A coexisting technology, BLE can be used for tracking, connectivity, and transmission of data though the precision is rather reduced compared to UWB with a lower speed of data transmission due to bandwidth limits. However, there is a holdback due to many developments conferred upon this technology which is yet to be ubiquitous on common devices compared to BLE, and Wi-Fi.

UWB is under a rapid development phase which would be ubiquitous in the future to cater to market trends, though the ecosystem is not in a move to rescind existing communication technologies. Regulations would enable manufacturers to make products standardized [6].

The wide adaptation of UWB shall ideally transform the technology into an essential service to be intact for various tasks in a large array of disciplines by integrating UWB devices to obtain data for tracking, localization, and efficient integration of measured



Figure 10.2: Tracking an ‘Airtag’ using the smartphone. Source: www.ndtv.com

data for better performance such as detection of a terminally ill patient using UWB pulses collected by sensors through movement detection, just to name a few [11].

Various organizations such as Samsung Electronics (South Korea), Sony (Japan), and Starix Technology (US) are trying to take grounds on UWB for a better ecosystem [9].

Omlox is an industry pioneer in clutching an approach on its API development for a real-time locating system as a commercial product using UWB through its interface and hubs which enables multiple devices from various manufacturers to connect to its interface which would map all the connected devices holistically and extract it to a single localization zone. The mechanism was enchanted by grasping the tracking of all the connected devices to a single platform. The mapping feature provided by OMLOX to bring all the devices to a one-stop shop provides the user to distinctly visualize the connected devices and their localization. OMLOX plays a vital role in regulation standardization with the respective authorities as a giant in the field of RTLS [15].

Figure 10.3 explains various levels of the OMLOX RTLS ecosystem. The middleware acts as the infrastructure ensuring interoperability between other technologies such as BLE, GPS, etc. It entails harmonizing data from local coordinates to global coordinates through the OMLOX engine.

10.4 Types of topologies available in UWB

Depending on the application, the phase or the time difference is used for localization. UWB technology mainly operates using four topologies. A distinct factor of the evaluation of ToF compared to other RF methods is the ignorance of reflections at the transceiver end thus providing better accuracy of the localization of a device with the use of the direct signal, eradicating misleading observances which leads to an incorrect conclusion of the original transmitter’s placement.

The concept of tags and anchors play roles in UWB localization where an anchor is the fixed infrastructure and the tag is in mobility. The device communicates with the

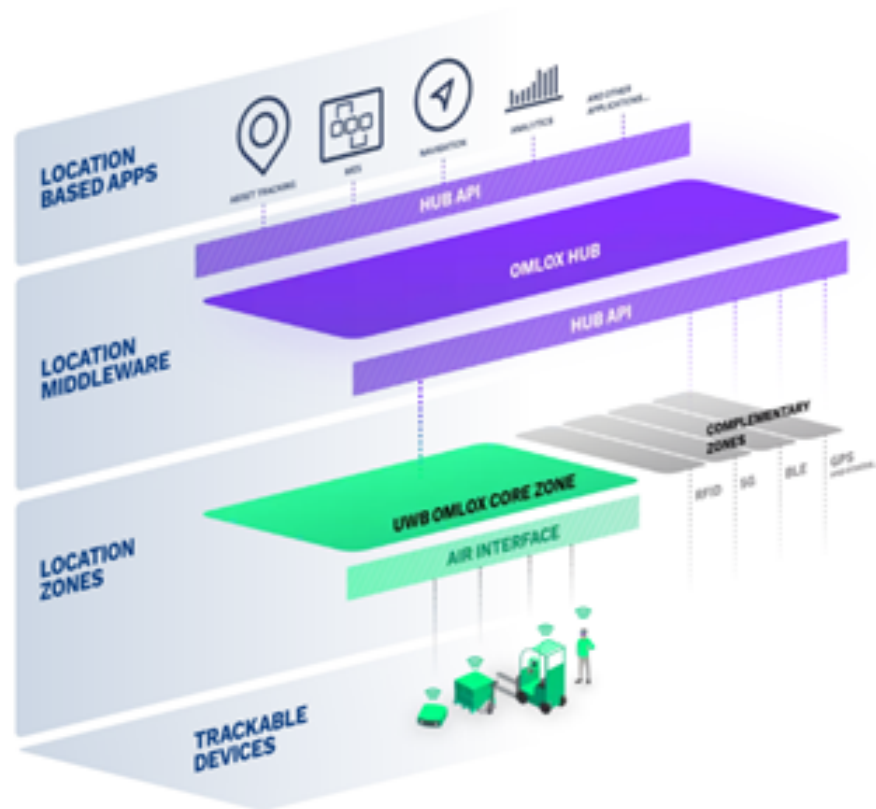


Figure 10.3: OMLOX API Configuration Stack. Source: www.omlox.com

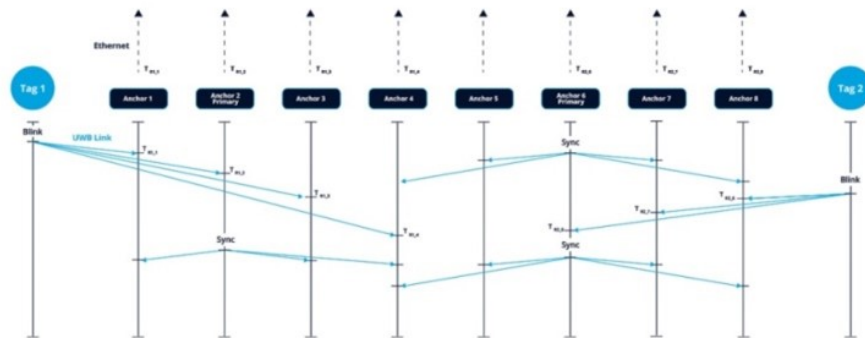
anchor continuously to localize though multiple anchors are required to extract a global localization relative to the coordinates where the anchors are placed.

10.4.1 TDoA or Time difference of arrival

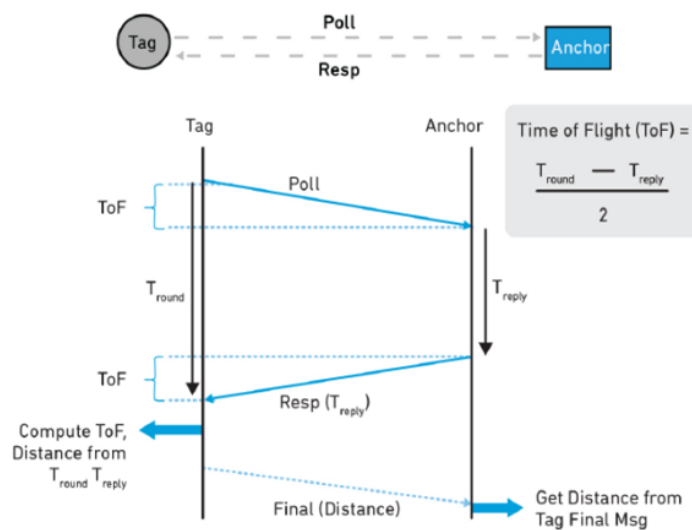
The Sensors and anchors attached to the system would continuously monitor the exchanged pulses within the system to trace devices. Accurate timing is required at every node to ensure the correct timestamp. Returned pulses would be analyzed by the location engine using multilateration to calculate its coordinates to determine the exact location of the device.

10.4.2 TWR or Two Way Ranging

Using this mechanism, multiple anchors would be communicating with the UWB device which is mobile or fixed to sense the distance between any given UWB device and the anchor. At any given frame of time, the proximity towards a specific anchor would initiate the calculation ranging to determine the distance of the device from the said anchor to calculate the distance.

Figure 10.4: Time Difference of Arrival. Source: www.qorvo.com

As shown in figure 10.5, a poll and response strategy will be deployed. The Tag initiates the poll, and the anchor would timestamp and respond to the tag. The tag has the information on the flight time and the processing time at the anchor end where a timestamp has been punched. This allows the tag to calculate the distance accurately. Multiple anchor coordination is yet possible though it may drain the power quickly as the overhead of the communication factor is present.

Figure 10.5: Two-way ranging. Source: www.qorvo.com

10.4.3 Reverse TDoA or Reverse Time Difference of Arrival

In this scenario, the mobile device computes the location with the pulses emitted periodically by an anchor. Similar to GPS multilateration can be used by the mobile device to compute its location.

An illustration of RTDoA is shown in figure 10.6. The tag in the middle tries to localize its position according to fixed anchors placed which emit UWB pulses continuously for global localization. Failure of an anchor may yet provide the capability of triangulation.

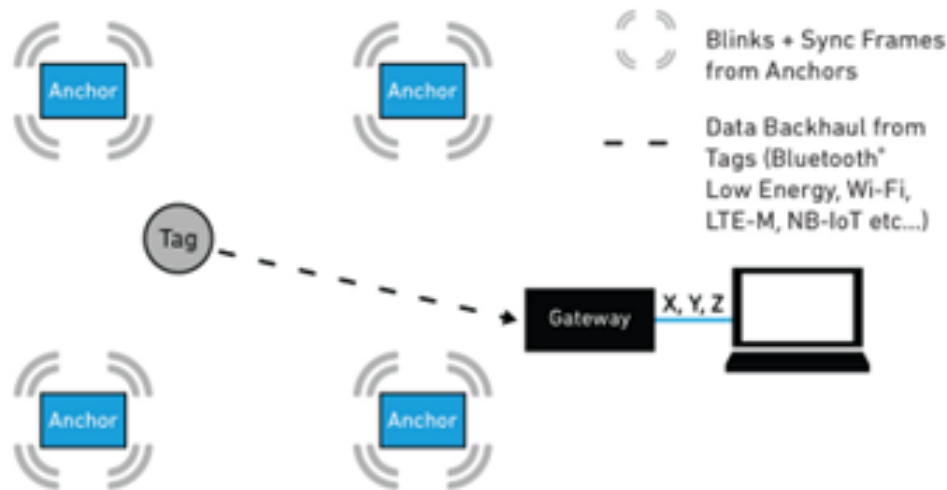


Figure 10.6: Reverse Time Difference of Arrival. Source: www.qorvo.com

10.4.4 Phase Difference of Arrival

A peer-to-peer mechanism enables a device to relatively know the position of another device thus directional calculations are made to poise the position. To calculate the phase differences at least one device must contain two antennas which can then calculate the phase difference of the receiving and the transmitted signal sequentially.

As shown in figure 10.7, the phase will be measured by both the antennas from the device which contains the structure. Subsequently, the calculation could be made to position the directional measures.

The topology illustrations above depict that the technical infrastructure requirements for UWB are straightforward. The utmost important equipment are the tags and the anchors where depending on the application there may be a coordinator and a system to analyze the signals.

As argued by [2], [3], and [6], the infrastructure layout of UWB is not extremely complicated compared to career-based signal processing. Since the technology uses low-powered pulses, the requirement of amplification would be a redundant process that is not required. Differentiation of entirety compared to career-based signal propagation, the pulse-based UWB would eradicate every inch of techniques used for the sustenance of a signal with the use of a career wave and modulation techniques engulfed to ensure the signal upholds its characteristics at the receiver's end. Thus neither mixers nor demodulation is required to extract the original signal back at the receiver's end.

This establishes the minimal infrastructure required to deploy UWB in contrast to conventional methods. Thus, the prime hardware requirement would be a UWB transmitter

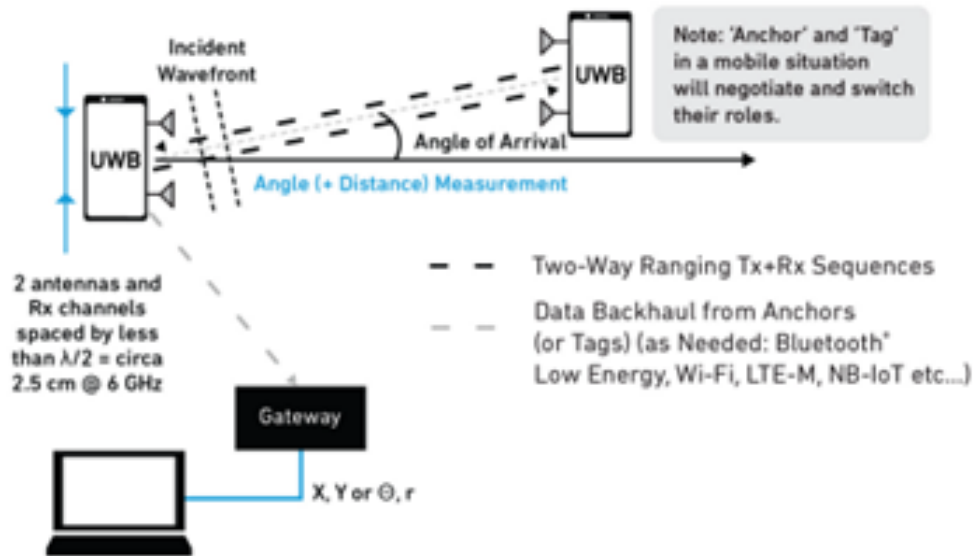


Figure 10.7: Phase Difference of Arrival. Source: www.qorvo.com

and a transceiver. The main components for the UWB to operate are an UWB-enabled transmitter which sends signals out such as a mobile device to locate and the anchors which act as transceivers to absorb the pulse emitted by the transmitter of the mobile device to calculate the proximity of it using a variety of methods.

10.5 Affiliate Organizational and Regulatory Authorities

According to [16], consortiums and organizations are striving to assist UWB evolution thus streamlining regulatory standards, which are namely;

- FiRa Consortium or Fine Range Consortium
- UWB Alliance
- Omlox
- Car Connectivity Consortium

As UWB is operated in the range of the unlicensed spectrum, regulatory bodies are stepping in to ensure the usage and rules are dignified on the utilization to ensure security, fair usage, and value. The requirement of a consortium becomes inevitable when technology is trying to break barriers to be ubiquitous.

Manufacturers exploit fresh technologies within their products for a competitive edge which escalated the need for a standard to ensure interoperability within the ecosystem. The ecosystem of UWB will only keep growing once the technology becomes pervasive

such as BLE and Wi-Fi. Hence the integration of a consortium would confine exploited technicality into a recognized and established standard. One may argue for the feasibility of making a global standard due to vivid jurisdiction in countries that may act as obstacles, though a few segregations could be drawn out for multiple standards which could be used in various demographic regions where there might be a conflict (e.g. Mobile bandwidth utilization in the USA and EU).

IEEE 802.15.4z was introduced as a standard for UWB which has outshined junctures since its evolution to absorb further security aspects such as encryption at the physical layer which is hard for an intruder to attack. Figure 10.8 is a representation of the respective responsibilities of IEEE and FiRa which their intervention is observed in the protocol stack for harmonizing UWB standards.

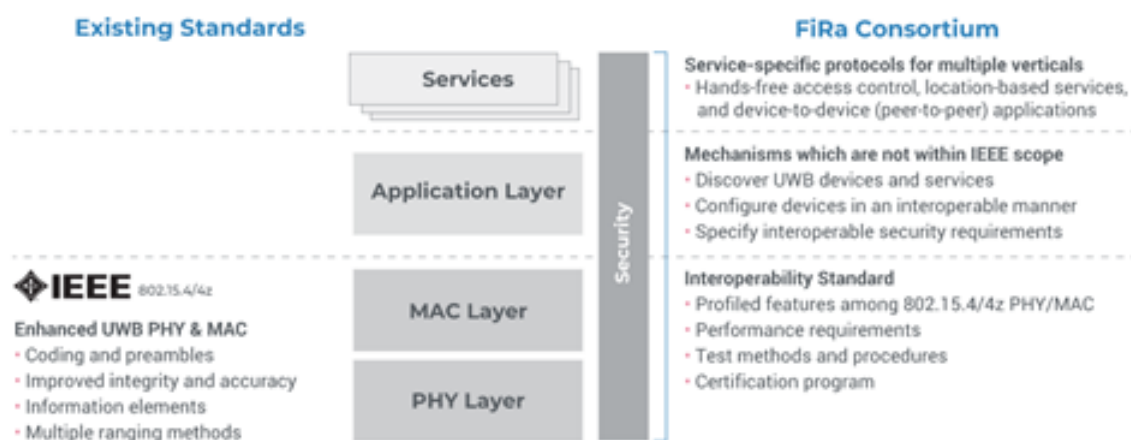


Figure 10.8: Classification and responsibilities of FiRa and IEEE relating to standard specification. Source: www.firaconsortium.org

UWB Allianz is playing a major role in adequate standardization of the technology for the growth of UWB by providing favorable regulation and spectrum management landscape which is on continuous optimization through collaboration. The important aspect is to build a smart ecosystem where UWB could reside within existing technologies which requires minimal interference [17].

Some of its work is related to advocating a new spectrum that aims to extend the current to the range of 12.4 GHz. Promoting the standard related to the initial IEEE specification is an ultimatum within its scope by extending any future work abiding by the amendments released.

It was argued as a pioneer in the field of providing real-time location solutions OMLOX is helping the respective organizations to build standards for seamless interaction and usability of the technology. As the first open locating standard the contribution from OMLOX is crucial for an open RTLS ecosystem due to their involvement with the regulatory bodies for standardization [15].

'CCC' or Car Connectivity Consortium is on the move to introduce a digital key that could be used to operate the vehicle using any smart device which may connect using UWB in

a secure method. For this purpose, CCC is collaborating with many stakeholders which would ultimately deliver and establish standards once pronounced for commercial use [16].

Thus respective organizations collaboratively engage in the evolution of UWB, standards, and sight for a broader ecosystem.

10.6 General application solutions visible in the industry using UWB

According to FiRa, UWB is used in many disciplines; noticeably UWB is getting deployed in the following commercial settings at present [10].

- Smart cities and mobility provide indoor navigation
- Smart building and construction for asset tracking
- The retail industry uses tracking the movement of the shoppers by evaluating foot-step movements
- Smart homes which provide the consumer to find something important such as the car key

Figure 10.9 depicts the navigational application embedded in the phone and a possible asset which is a suitcase of belongings. The Airtag would be continuously traced to identify the current location for easy retrieval. Once the Airtag loses connectivity with the respective device of the owner, the tag would use the Apple ecosystem as a crowdsourcing resource to trace the bag which entails privacy concerns.

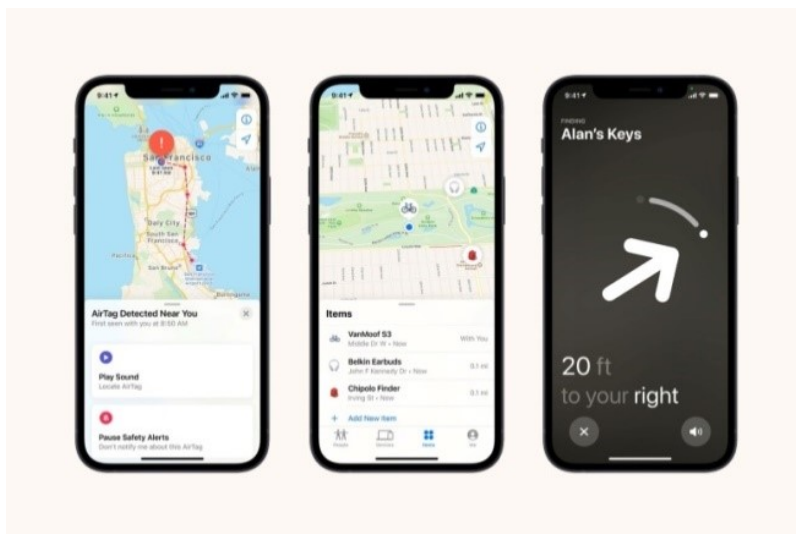


Figure 10.9: Precision Proximity Guidance. Source: www.techspot.com



Figure 10.10: Airtag on a suitcase. Source: www.executivetaveller.com

10.6.1 A Glimpse of UWB Usage in Apple Air Tags

Apple ‘Airtags’ work by connecting air tags to objects of desire of the consumer thus the tag could be associated with the phone and traced continuously with the pulses emitted by the ‘Airtag’ to the corresponding device of the owner which enables the consumer to keep track of the object if the person loses the sight otherwise. The application on the phone would guide to the closest proximity of the object where the tag manages the precision up to centimeter level [7]. According to Heinrich [12], BLE is used for tracking and once the tag is located the UWB chip embedded within the phone tackles precision localization thus there is a transition of technologies working collectively, Said functionality is considered useful though according to Ahmed [9], the aspect of crowdsourcing has raised ethical issues due to resource usage without consent.

Some controversial privacy and user trust concerns are tabled in the arena thus the following questions must be answered to ensure ethics and user trust. A set of ethical concerns are identified which will be discussed in a later chapter.

Similar to Airtag, Samsung has introduced its Samsung ‘Smart Tag’ and then another by TILE, a company that is predominantly in the business of tracking devices. Each of them has its drawbacks of tracing as per the discussion of Fowler [13] which will be elaborated in the latter sections relating to privacy concerns. However ‘Tile’ uses BLE for localization and tracking instead of UWB, unlike the rest of the mentioned.

10.7 UWB applications in the healthcare sector

Gosh argued healthcare sector has attained tremendous benefits with the use of UWB in its respective field [5]. According to Alghanimi [25], medical applications on the ultra-wideband could be assorted into the following regions but not limited to;

- Cancer and Hemorrhage Detection in Organs

- Position and Localization
- Non-contacting medical imaging
- Vitals monitoring
- Heartbeat and Lung movement detection
- Blood glucose measurements
- Heart volume detection vascular and vascular pressure measurements

10.7.1 Prior research and related work done in the field of medicine

Staderini [1], probed into the possibilities of integrating UWB for medical applications for dedicated use of accuracy with the use of electromagnetic pulses emitted by UWB. This radar technology was used to measure the movements of the cardiac wall by extracting the reflection magnitudes of the returned pulse. This was possible due to the impedance difference between the blood and the cardiac muscle. This mechanism could be used for other organs such as the movement of the lungs compared to its boundary interfaces which could be air, blood vessels, tissues, or another prime organ. This would enable practitioners to monitor the cardiovascular system of the patient and employ observation to detect contractions and respiratory movements. The added benefit would be the zero impact made by the wearables of the patient as the magnetic pulses would simply move through with no hindrance of effectiveness. Staderini, S.M. ventured to detect movements of various vital organs in their respective urgency with a higher accuracy where typically used ultrasound could be replaced which needs to be physically intact with the surface area in a pregnant woman. Vital data of the entire development has been taken care of by extracting measurements of both the mother and the fetus accurately.

UWB proves to be compatible due to its nature of non-harmful emissions to the patient and the baby at critical moments in the delicate setting. Similarly, low radiation is harmless even during an examination at a chronic stage due to UWB exposure [20]. This may subsequently establish the fact that the emitting equipment would carry a long-life cycle until the battery is replaced due to minimal power consumption.

Bharadwaj [4] was interested in precise motion tracking of limbs and human subjects using impulse radio ultra-wideband. This 3D movement tracking was accomplished with the use of compact and cost-effective body-worn antennas. Various parameters were used in the system to calculate the locality with high precision by using path loss magnitude, number of multipath components, RMS delay spread, signal amplitude, and Kurtosis. The fidelity was scrutinized by choosing different places in the human body for higher localization accuracy. A mechanism as such would ideally suit to detect movements of a patient remotely where instant actions need to be taken due to the nature of the medical condition where responsible authorities could be alerted during any adversarial or abnormal reading observation from the UWB transmitters to initiate swift actions. Deep penetration characteristic is quite useful which has seamless integration with MEMS and Inertial units for tightly coupled localization accuracy.

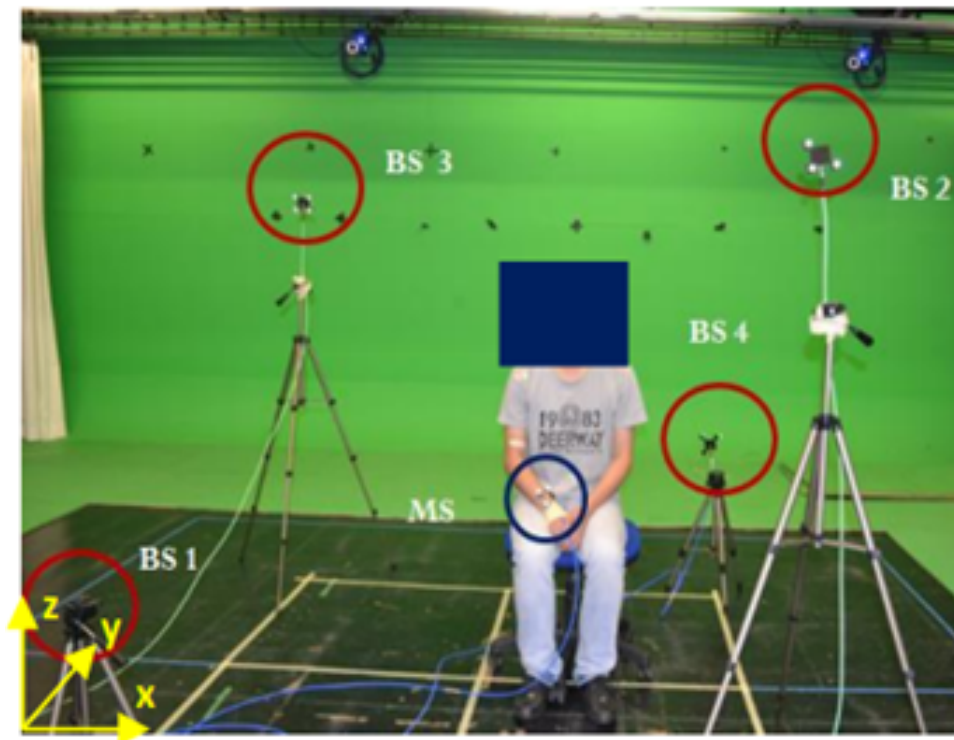


Figure 10.11: Localization of limb movements. Source: Impulse Radio WB [4]

In Figure 10.11, base stations would act as anchors to collect the movements of the body through the antennas placed on respective areas which are then sent to a vector network analyzer to localize the object.

The concept used by Bharadwaj [13] has been exploited by much other research mainly for localization in a diversity of medicinal application scenarios. During remote patient monitoring collected data could be transmitted to the coordinator via IoT system for analysis [5].

An interesting concept has been introduced by Rahman [8] to monitor the healthcare of children and adults with the use of WBAN. This would enable a whole combination of designated sensors to constantly transmit vitals of the vulnerable individuals to be monitored which then could be alerted during an adversarial situation or to observe the quality of life by tracking characteristic changes during a routine lifestyle. This enables the subject to carry on their lifestyles with no interruptions from outside though the situation would be continuously observed and monitored by a responsible third party for any possible adversarial detection to be addressed. Accuracy of the measures in this situation is quite important as mentioned formally hence UWB can play a major role in initiating this. It is important to note that synchronization of all the anchors and the device with the correct time stamp is crucial for an accurate reading.

However, the research was more focused on available wireless technologies which can be used for healthcare rather than concentrating purely on UWB but an elicitation of available methods and constraints.

A fine set of sensors could be used in any case of data collection during a UWB implementation to transmit the data. These may be but not limited to the following:

- Inertial motion sensors: to absorb the movement of the body
- Bio Electric sensors: Electric signal generation due to biological activities such as ECG
- Temperature Sensors
- Optical Sensors to evaluate the oxygen levels

Karad [11] proposed a prototype of an idea for a flexible UWB antenna which was manufactured using a substrate that would not irritate the human skin once placed.

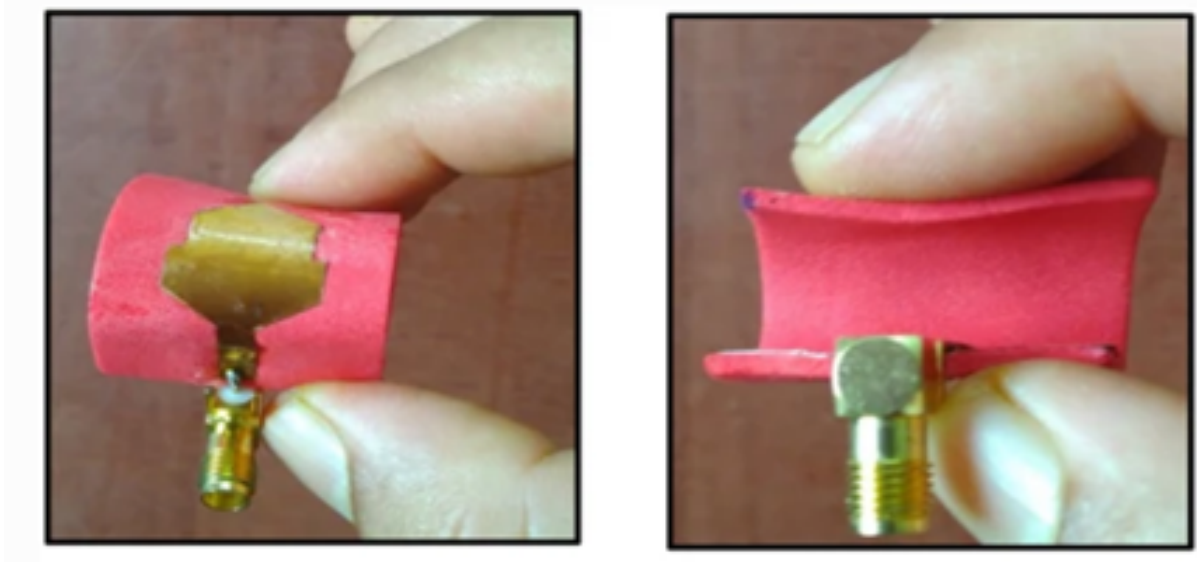


Figure 10.12: UWB transmitter placement on a flexible substrate [11]

The flower bud-type antenna could be easily positioned in the desired location. The chemical composition of the substrate and meta characteristics such as heat insulation, better cushioning, shock absorption, and water resistance make the prototype convenient to use as a wearable for health applications. Karad's concept brings light into the field of user experience of the UWB device which the consumer would be carrying on them. The aspects of comfort considerations play a vital role in the user's desirability to carry the UWB device or resistance due to discomfort. The research proved useful to combine the heuristics of UX with the integration of chemistry to decide upon the substrate.

A UWB baby monitor is another viable product deployed for healthcare monitoring with the use of UWB. Ziganshin [27] conceptualized UWB-embedded baby monitors which can be used to prevent sudden infant death syndrome. This was quite efficient to combat sleep apnea which could be detrimental for an infant as a conventional baby monitor would not accurately measure delicate pulses which occur in the rhythm of the respiratory system to identify violations. High precision of UWB proves to be extremely useful in this situation

to overcome the concern by alarming the parent when an abnormality is detected. The reliability of this system comprises two independent-spaced receiving channels due to lower reflection.

10.7.2 Applications using various topologies of UWB in Healthcare

TWR methodology can be used to monitor Ataxia patients to detect walking abnormalities. Zilani [24] argued that ‘Ataxia’ patients lack proper balance coordination in the body which impacts mobility. Walking movement measures can extract using tags. The Anchor can calculate the accurate localization by taking the time of flight in to measure and then multiplying by the speed of light. Through the placement of 3 anchors at the same height the multi-lateralization is performed to acquire the location where a SVM is used to classify different movement patterns.

Zhang [23] proposed a method to facilitate the localization of elderly people through Phase Difference of Arrival. The prominence of this topology came to light due to the line of sight (LOS) concern which shadows at homes. As mentioned in Chapter 4, when obstacles prevail in the range of the UWB transmission, multipath propagation interference impacts the accuracy of the signal. Insertion of the PDoA chip allows the base station to coarsely estimate the location of the elderly which provides a glimpse of the LOS/NLOS factor where the compensation is only required for NLOS.

Livinsa [22] articulated biomedical research using TDoA for patient localization and monitoring due to low signal strength indoors. The rich factor of TDoA is that no synchronization is required between the transceiver and the transmitter. The time stamping mechanism used ensures higher accuracy for localization. EKF algorithm can be used to further enhance this method for the subject of mobility.

Finally, AoA is a subset of PDoA which requires two antennas and thus can deem quite costly but the benefit relies on the non-synchronization of time stamps though the angles between the transmitters and the anchor must be measured accurately, a topology that can be used to accurately localize.

10.8 Privacy and User Trust Concerns of UWB

As mentioned formally, the functionality of UWB somewhat raises privacy concerns. It is noted that these concerns are mainly based on the application of UWB rather than the intrinsics of the technology itself. Related issues may arise due to the controversy of permission given to trace an activity without the knowledge of the subject [13].

A widely spoken scenario in this regard is tracking on mobile devices where especially mobile phone customers were complainant against ongoing activity tracking with the tracking symbol observed on their phones which depicts that ‘UWB’ is active though there is no activity performed by themselves which led to a concern of unauthorized usage of a third party device without the consent, breaching the privacy. The situation may arise

due to various reasons whether it could be a bug in the system, or some other applications are already intact with UWB for their persistence [28].

There can be severe consequences due to any intruder intervention to track data of a device without the knowledge of the owner and this is not purely aligned on UWB but on any tracing mechanism. There could be detrimental impacts through the possibility of stalking essentially which takes a mile closer with the use of UWB due to higher precision.

UWB can be used both as an infrastructure or an application and each of them has its benefits and consequences though most of the privacy concerns are escalated due to the localization accuracy due to applications rather than the infrastructure.

There are instances where a health device would be attached to the body to continuously transmit information relating to the vitals of a subject such as the blood pressure, pulse, or heart rate. At a glance, this may not look intrusive. In a coordinated environment where UWB transceivers are placed for data retrieval directly onsite or through IoT connectivity may not be detrimental to the subject rather the collection of data would assist Medicare centers for early diagnosis of any devastating situation of a patient through continuous monitoring.

However, there is a possibility that the data acquired through tracking may go to third-party vendors which might be stolen or deliberately sold to an entity which again lays a heavy weight on the coordinating organization and the subject itself whether any consent has been provided to manipulate the data for unauthorized exposure.

Especially in the field of healthcare, data collected by subjects are quite vital for many business-oriented tasks. It may be to manufacture a new drug, or equipment to better suit to navigate through the task, which may lead to market research for a new product to be introduced for the mainstream market for financial gains.

The malicious intent of research without consent paves the way to a new product in the market with an utter monopoly that would harness a higher demand for the said deliverable which is absolutely niche and vital as the strength of the UWB is quite established when it comes to performance and benefits. An ethical issue can be raised in this situation if the subject's consent has not been provided to perform a task as such where holistically the situation proves useful for the subject as well as the entity [25].

10.8.1 Situational analysis of tracking devices during its usage related to privacy - UWB in Mobile Devices

Prime privacy concern escalates with the possibility of tracking another person by simply setting a tracker within or embedding to another's possessions. This would allow a stalker to track the mobility of the item which would be ideally the movement of the person.

UWB tracker manufacturers have been thoughtful towards this issue and resolutions are implemented. Apple phone managed to escalate an alert stating that "An 'Airtag' is moving with you" which would notify the person who has got no knowledge of an 'Airtag' around the user via a beep. Heinrich classified this as 'Offline Tracking' [12][13]. Tile made

the situation better by introducing a scan button to identify any unknown tags following the person though the scan button needed to be pressed for the device to search for any tag around. Software compatibility issues of the Smart tag raised further questions about the operational mechanisms of air tags as the ecosystem and the compatibility issues arise depending on the phone and the applications [13].

Stating the above, all the solutions which were provided had a further escalation of empirical usage such as false alarms, the duration to be notified, automated notification periods, Manual scanning requirements to identify a tracker, compatibility with applications and manufacturers etc.

When relating the topic of UWB, the scenario of the usage of tracker in mobile phones would provide a better glimpse of how the system works due to the pervasiveness of the technology and the hype it has created amongst the usage of iPhone or Android phones with the compatibility UWB feature has long broken out as a commercial product.

10.9 Security of UWB technology

An important mechanism that was introduced through UWB is the localization via the calculation of Time-of-Flight Difference. According to [18], the IEEE 802.15.4z standard operates in two different modes where the packet arrangements would differ significantly. Two operational modes are LRP (Low Power Pulse) and HRP (High power pulse).

In a topology such as 'Time of Flight' the estimation would be made via the calculation of the correct timing and the speed used which provides the distance. An external intruder can manipulate these signals and control distance estimation between legitimate entities. It is quite evident that UWB is shielded by relay attacks by definition as the time-of-flight mismatch would easily prove a presence of an intruder which the access could be denied though this can be manipulated by the adversary. These mechanisms are quite useful for passive entry and start systems in vehicles and for digital locks in households to enter.

According to Kshetrimayum, 2009 the EIRP or Effective Isotropic Radiated Power has been defined as the highest measure of the signal which prevails in coexistence with the other devices in the same frequency region and the less emission of UWB to avoid detection and jamming. This is primly due to UWB operating power of -41dBm which alleviates interferences with others and also reduces spoofing attacks.

Having multiple channels for operation due to the large bandwidth pushes the adversary to the edge for a broader scan of the spectrum to deploy the attack. The speed of the pulses sent in intervals makes the adversary work hard for it though a chain of security vulnerabilities could be drawn through hacking into a system via tracing or crowdsourcing.

Following are some attacks which may create security vulnerabilities in UWB:

- Cicada Attack: In their attack, an adversary transmits uniformly spaced ultra-wideband pulses during the transmission of the legitimate preamble. The presence of the attack signal degrades the performance of the receivers by undermining

leading-edge detection, resulting in distance reduction [19]. This may happen due to the difficulty of identifying the signal with the direct path which may not be the strongest. In this scenario the presence of the attack signal degrades the performance of the receivers by undermining leading-edge detection, resulting in the distance reduction.

- **ED/LC Attack:** The attack works by exploiting the predictability of the (inner) signal structure of a symbol where the adversary learns the packet/symbol values early and commits them late to fool receivers about the signal arrival time. Though the signal is encrypted with a challenge or a nonce, the adversary can reduce the distance through an attack as he knows the signal.
- **Cicada ++ Attack:** An attacker probes into a MITM attack by sniffing the legitimate signals. The adversary amplifies the signal which creates the highest correlation peak. Synchronization with the legitimate pulses thereon would provide a rough estimate to the infiltrator about the signal behavior thereby intercepting it.
- **Adaptive Injection Attack:** The adversary injects a peak before the legitimate arrival time thus the attacker gains control of the imputed peak to intercept the signal or to deploy a relay attack.

There are solutions and mechanisms introduced to optimize the security of UWB and namely:

- Noise Floor Estimation
- Peak Detection Thresholds
- STS Peak Candidates
- Leading Edge Detection
- Ranging Error

Moreover, encryption would be done in the physical layer where the signal cannot be sniffed in the middle which may have otherwise provided the intruder the opportunity to manipulate the data and compute approximate anchor positions to initiate relay attacks.

Likewise, this encryption alleviates the probability of replay attacks as the adversary is capable of obtaining information about the anchors through the timestamps by interception therefore a replay attack could be initiated with customized characteristics conferring to the trespasser.

Similarly, embedding systems such as firewalls, intruder detection systems, and SSH tunneling with packet synchronization for data encryption would also be classified as measured to be taken to ameliorate security vulnerabilities that may occur in a situation such as geofencing.

10.10 Conclusion

Features of UWB have drawn broader attention in the market by manufacturers in vivid disciplines purely based on the magnitude of the beneficial factors it entails. Comparison of many wireless technologies and their usage proved the advantageous influences of UWB which were lagging behind the rest of the pervasive technologies due to the innate characteristics of UWB as mentioned in sections 2 [1] and [14].

As stated above, it's imminent that UWB will be ubiquitous momentarily considering the momentum and the support it gains from various consortiums and organizations. Rather than monopolizing the wireless technologies, UWB is predicted to co-exist within the ecosystem of already prevailing mechanisms such as BLE and Wi-Fi and many more. However, due to the higher-level accuracy provided for localization, security, speed, and penetration capabilities, UWB is in a rapid development phase for domain-specific applications which awaits a rollout in various disciplines in construction, healthcare, and communication.

At present UWB has been exploited in various segments whether it could be in asset tracking, smart houses, or in the healthcare sector though the most common hype was propounded by 'Airtags' and later mobile tracking systems. User privacy and ethical concerns are quite spread out in the arena to be debated despite the gains.

Ultra-precision to a 'cm' level of accuracy of UWB is quite useful in many applications though there is a tradeoff to be evaluated regarding the elevated privacy concerns due to crowdsourcing mechanisms which manufacturers are trying to resolve concerns.

Considering the security methods infused in UWB, the technology can be classified as quite secure compared to the others with added benefits such as power saving and non-detrimental effects on humans due to radiation. Singh argued however, drawbacks could be identified relating to possible attacks to be deployed on the user where secure mechanisms are in place and development as a mission-impossible task to eliminate the possibility of attacks.

As stated above, and through a holistic and precise evaluation of the capabilities of UWB, it is eminent for the technology to break barriers and be ubiquitous where consortiums such as FiRa would abide to ensure the standards to be established for the technology to sustain thus to address voices raised through its development. The availability of this technology is vital for the development of novel mechanisms to act as a service to be provided the mean of a device or established infrastructure.

Bibliography

- [1] E.M. Staderini: *UWB radars in medicine*, IEEE Aerospace and electronic systems magazine, 17(1), pp.13-18, 2002.
- [2] M.Z. Win, D. Dardari, A.F Molisch, W. Wiesbeck, and W.J Zhang: *History and applications of UWB*, Institute of Electrical and Electronics Engineers, 2009.
- [3] R.S. Kshetrimayum: *An introduction to UWB communication systems*, IEEE Potentials, 28(2), pp.9-13, 2009.
- [4] R. Bharadwaj, S. Swaisaenyakorn, C.G. Parini, J.C. Batchelor, and A. Alomainy: *Impulse radio ultra-wideband communications for localization and tracking of the human body and limbs movement for healthcare applications*, IEEE Transactions on Antennas and Propagation, 65(12), pp.7298-7309, 2017.
- [5] D. Ghosh, and P.K. Sahu, P: *UWB in healthcare*. In *2016 International conference on electromagnetics in advanced applications (ICEAA)*, pp. 679-682. IEEE, 2016 September.
- [6] *IEEE Standard for Low-Rate Wireless Networks–Amendment 1: Enhanced Ultra Wideband (UWB) Physical Layers (PHYs) and Associated Ranging Techniques*, IEEE Std 802.15.4z-2020 (Amendment to IEEE Std 802.15.4-2020), vol., no., pp.1-174, 25 Aug. 2020, doi: 10.1109/IEEESTD.2020.9179124.
- [7] C. Boyd: *What is an Airtags precision finding? Everything you need to know!*, Payette Forward, Published in 2021, Last accessed on 13/03/2023. <https://www.payetteforward.com/what-is-airtags-precision-finding/>.
- [8] Aqeel-ur-Rehman, I. Abdullah and S. U. Rehman: *Wireless Body Area Networks for Child Healthcare Monitoring: A Review*, 2018 IEEE 21st International Multi-Topic Conference (INMIC), Karachi, Pakistan, 2018, pp. 1-8, doi: 10.1109/INMIC.2018.8595655.
- [9] A. Ahmed: *Privacy issues of mobile phone companies' usage of Ultra-Wideband (UWB) technology: Analyzing the use of UWB in mobile phones from a multi-actor perspective, magnifying privacy concerns and formulating guidelines*, 2021.
- [10] *Our Vision*, FiRa Consortium, Published in 2023, Last accessed on 13/03/2023. <https://www.firaconsortium.org/>.

- [11] K.V. Karad, and V.S. Hendre: *A flower bud-shaped flexible UWB antenna for health-care applications*, EURASIP Journal on Wireless Communications and Networking, 2023(1), pp.1-18, 2023.
- [12] A.Heinrich, M. Stute, T. Kornhuber, and M. Hollick: *Who can devices? security and privacy of apple's crowd-sourced Bluetooth location tracking system*, Proceedings on Privacy Enhancing Technologies, 2021(3), pp.227-245, 2021.
- [13] G. Fowler: *Am I being tracked? Anti-stalking tech from Apple. Tile falls short*, The Washington Post, Published in 2022, Last accessed on 13/03/2023. <https://www.washingtonpost.com/technology/2022/03/31/airtags-stalking/>.
- [14] T. W. Rondeau, M. F. D'Souza and D. G. Sweeney: *Residential microwave oven interference on Bluetooth data performance*, IEEE Transactions on Consumer Electronics, vol. 50, no. 3, pp. 856-863, Aug. 2004, doi: 10.1109/TCE.2004.1341691.
- [15] *The open locating standard*, OMLOX, Published in 2022, Last accessed on 25/04/2023. <https://omlox.com/>.
- [16] *Getting back to basics with Ultra Wide Band*, Last accessed on 25/04/2023. <https://qorvo.com/>.
- [17] *UWB Allianz Homepage*, UWB Allianz, Last accessed on 25/04/2023. <https://uwballiance.org/>.
- [18] M. Singh, M. Roeschlin, E. Zalzal, P.Leu, S. Capkun: *Security analysis of IEEE 802.15.4z/HRP UWB time-of-flight distance measurement*, pages 227-237. <https://doi.org/10.1145/3448300.3467831>.
- [19] B. Toulas: *RTLS systems vulnerable to MiTM attacks, location manipulation*, Last accessed on 02/05/2023. <https://www.bleepingcomputer.com/news/security/rtls-systems-vulnerable-to-mitm-attacks-location-manipulation>.
- [20] J. Pan: *Medical applications on Ultra-Wideband*, Published in 2008. <http://www.cse.wustl.edu/~jain/cse574-08/index.html>.
- [21] A. Mallat, J. Louveaux and L. Vandendorpe: *UWB based positioning: Cramer Rao bound for Angle Of Arrival and comparison with Time Of Arrival*, 2006 Symposium on Communications and Vehicular Technology, Liege, Belgium, 2006, pp. 65-68, doi: 10.1109/SCVT.2006.334374.
- [22] Z.M. Livinsa, and S. Jayashri: *Biomedical Research*, Special Issue: S400-S406, 2016.
- [23] Y. Zhang and L. Duan: *Toward Elderly Care: A Phase-Difference-of-Arrival Assisted Ultra-Wideband Positioning Method in Smart Home*, IEEE Access, vol. 8, pp. 139387-139395, 2020, doi: 10.1109/ACCESS.2020.3012717.
- [24] T.A. Zilani, F. Al-Turjman, M.B. Khan, N. Zhao, and X.Yang: *Monitoring movements of ataxia patient by using UWB technology*, Sensors, 20(3), p.931, 2020.

- [25] *How predictive analytics can boost product development*, McKinsey, Last accessed on 25/04/2023. <https://www.mckinsey.com/industries/technology-media-and-telecommunications/our-insights/how-predictive-analytics-can-boost-product-development>.
- [26] A.H. Alghanimi: *Medical Application of Ultra-Wideband Technology*, Innovations in Ultra-Wideband Technologies, Published on 28/07/2021.
- [27] E. G. Ziganshin, M. A. Numerov, and S. A. Vygolov: *UWB baby monitor*, 2010 5th International Conference on Ultrawideband and Ultrashort Impulse Signals, IEEE, 2010.
- [28] *Apple AirTag*, Apple, Last accessed on 20/05/2023. <https://www.apple.com/airtag/>.

