Chapter 1

Introduction

1.1 Motivation

In recent years, cyber insurance has emerged as a new form of insurance. A customer seeking cyber insurance is usually looking for protection against a variety of cyber risks, that might include hacking incidents, data leaks, IT service outages, and the like. Asymmetry in information is one of the major problems in the insurance industry. As such, insurers have significantly less information about insured objects (e.g. a vehicle or a home) than their customers. This complicates risk assessment and insurance pricing. To counteract this, insurers would usually hand out questionnaires to their customers to gain more information about the underlying objects. In terms of cyber insurance however, this is not always feasible. Clients are generally reluctant to fully disclose details of their IT infrastructure, and security policies. Customers likely are concerned, that honest answers about poor IT-security practices might be used to discriminate against them, either at the time of insurance pricing or at a potential future claim settlement.

The recent advances in TEEs and privacy-preserving machine learning offer new avenues for dealing with the aforementioned problem of information asymmetry. A hardware-protected enclave (e.g. using Intel Software Guard Extensions (SGX)) could be leveraged to collect private customer questionnaires, and subsequently train a machine leaning model on the data in a privacy-preserving fashion. In order not to leak any secret information about the individual participants, techniques like DP can be used. In this way, the insurance company is able to build a useful aggregate model, while protecting the privacy of each individual customer at the same time. According to recent studies, with such a privacy-preserving system in place, users would be more willing to provide private data to data collectors.

1.2 Problem overview

Continuing with the insurance use case, there are two main parties involved, each with different demands: (i) Insurance customers need data privacy under all circumstances. Even a potentially malicious insider at the insurance company must not be able to leak secret data. (ii) The insurance is looking for good results from the resulting model. Customers or other external parties should not be able to distort the output model or render it useless through targeted interaction with the enclave. There is a number of things that could go wrong, which would lead to these requirements being violated. First, even though the training process takes place inside a TEE, leakage of secret information can occur. It is a known issues that TEEs are susceptible to side-channel attacks. Second, even if all side-channels during the DP-GBDT training are eliminated, developing a secure deployment strategy is a challenging task: How should data be moved into and out of the enclave in a safe manner? How can persistent state be added to the enclave in order to counteract forking/interleaving/rollback attacks? How can the system cope with data loss and other human or machine related faults?

Goals (i) *Efficient algorithm*: First of all, this requires building a standalone DP-GBDT implementation from scratch. The algorithm should be bug-free and able to produce accurate performance measurements. Ultimately, the algorithm should deliver decent results on a small privacy budget. In other words, the noise added through differential privacy should not completely erase the output model's expressive power. Additionally, even though training will likely not be performed very frequently, its runtime should be reasonable. (ii) *Side-channel resistance*: For secure execution inside the TEE the DP-GBDT implementation must be adequately hardened. The goal is to prevent all leakage through "digital side-channels". This notion sums up all side-channels that carry information over discrete bits. Examples are the memory access trace, control flow, and time. We do not need to get rid of side-channel leakage entirely. We must eliminate just enough leakage to achieve ε-differential privacy. (iii) *Secure deployment*: We further aim to explore the challenges of real-world secure deployment of the system. This includes: Privacy attacks through enclave state rollbacks or fork attacks, data collection and provisioning over an extended time period, ability to continuously improve the previous model with newly acquired data, resilience and fault tolerance in the case of human or machine errors.

1.3 Approach

Two prior works form the basis and starting point for this thesis. With DPBoost, Li et al. ([45], 2020) provide the theory behind DP-GBDT learning. Moreover, Théo Giovanna built a first Python implementation of said algorithm during the course of his master's thesis (submitted Feb. 2021). With the overlying goal of performing DP-GBDT inside an SGX enclave, a new C++ implementation had to be created. This lead to the discovery of several bugs and to substantial runtime improvements. The resulting code was afterwards ported into an SGX enclave. Subsequently, manual side-channel hardening on source code level was conducted. The process differs from traditional hardening to some extent: We do not try to eliminate every single bit of leakage. Instead, we remove just enough leakage at the right places to achieve ε-differential privacy. This is much more effective than tool assisted hardening as we leverage our knowledge of the DP-GBDT algorithm. Further, several data-oblivious and constant-time building blocks are created to replace common and reoccurring operations. We assume that the underlying hardware of the enclave setup is not compromised, patched, and works as expected. Denial of service attacks are also out of scope. The prediction accuracy of the algorithm is evaluated on four real-world UCI standard datasets. Moreover, we provide runtime measurements that show the impact of our hardening methods. Finally we explore possibilities for secure deployment of our enclave setup. With the help of secure monotonic counters, persistent state is achieved. We highlight the challenges and propose solutions for the enclave initialisation, data collection, training, enclave replication and migration.

Do we need some results teasers here? kari:yes

1.4 Contribution

The main contributions of this thesis are:

- Build a C++ DP-GBDT implementation that runs inside an SGX enclave
- Harden the implementation against side-channels
- Evaluate the accuracy and runtime of the algorithm
- Explore possibilities for secure deployment

There are further some smaller contributions that developed along the path: (i) Previous results and implementations of the DP-GBDT algorithm we are using turned out to have several flaws. To our knowledge, this is first time detailed and correct performance results of this algorithm have been generated. (ii) The underlying codebase was designed with a focus on usability and extensibility for future experimentation. (iii) This thesis offers some guidelines on manual (source code level) hardening of a tree based machine learning algorithm. (iv) Similarly, we offer guidance on how port such an algorithm into an enclave. What kind of code changes are necessary and how to divide code into inside/outside of the enclave.

Chapter 8

Related Work

Privacy-preserving machine learning is not a novel concept. A number of proposals exist that combine decision tree based classification of private data in a multi-party setting (e.g. [23, 46, 24]). One possible approach is to leverage homomorphic encryption schemes [4, 66]. Another approach is the use of Differential Privacy (DP), as introduced by Dwork et al. [29]. It can be argued, that this is the only mathematically rigorous definition of privacy in the context of machine learning and big data analysis. Through extensive research and growing industry acceptance, DP has become the standard of privacy over the past decade [6]. Multiple DP-GBDT solutions [45, 1, 47, 65] have been proposed since then. The combination of DP-GBDT with SGX enclaves has not been as thoroughly researched however. To our knowledge, there are two works that take a relatively similar approach to this thesis:

Allen et al. (2019) "*An Algorithmic Framework For Differentially Private Data Analysis on Trusted Processors*" [6]. Their high-level goal and architecture are the same as ours: Run DP algorithms inside SGX enclaves and eliminate leakage. Specifically, this work proposes a mathematical model for designing DP algorithms in TEE-based setting. They assume that the leakage only consists of (i) the output model and (ii) memory access trace. In this setting they ensure that DP guarantees hold for three selected algorithms. Decision trees are not among them. Further their focus lies more on the algorithmic side, which means they don't consider an entire system with data provisioning from users, disk as persistent storage, and so on.

Law et al. (2020) "*Secure collaborative training and inference for XGBoost*" [44]. The authors present a privacy-preserving system for multiparty training and inference of XGBoost [19] (efficient, open-source GBDT library) models. Their goal is to protect the privacy of each party's data as well as the integrity of the computation with SGX enclaves. However, they only consider side-channel leakage through memory access patterns. For this purpose, multiple data-oblivious building blocks for GBDT are created. Another difference to our approach is that no DP mechanisms are used. In other words, they aim for complete obliviousness of the entire algorithm, while we only selectively harden certain areas to achieve DP.

Chapter 9

Conclusion

In this thesis we present the implementation and hardening process of DP-GBDT, a relatively new variation of privacy preserving machine learning with promising applications. Compared to predecessor implementations, runtime performance was drastically improved and multiple bugs were identified and removed. The implementation was further ported into an SGX enclave and hardened to achieve ε-differential privacy. Affected code sections were protected from leaking secrets through *digital side-channels*, a notion which sums up all side-channels that carry information over discrete bits (such as the memory access trace, control flow or time). We saw that eliminating 100% of leakage is hard to achieve due to inherently non-constant-time hardware instructions such as several floating point arithmetic operations. The entire implementation and hardening process is captured in detail and should offer some guidelines for future work. The prediction accuracy of the DP-GBDT algorithm was evaluated on four real-world UCI standard datasets. Although we are not far from achieving good forecasting results with smaller privacy budgets, further research is required. Runtime performance of the training process was increased by over 500x compared to the predecessor implementation. This offered some cushion for the hardening overhead that subsequently added. The execution time of the fully hardened enclave implementation is well into the acceptable range for an algorithm that is not supposed to be run very frequently. In addition, we proposed a secure real-world deployment scheme of the DP-GBDT enclave setup in the insurance use case. Both the insurance and its customers are sufficiently protected from incidents covered by our threat model. At the same time, we placed emphasis on general usability and fault tolerance. Ultimately, we further reduced the gap between privacy preserving state-of-the-art machine learning frameworks and their real-world application.

9.1 Future Work

While this work shows encouraging results, there are still several paths to be explored in future work. On the theoretical side, further thought has to be put into making the algorithm more efficient in terms of privacy budget. On the practical side there is a broad selection of things that would add value to the project:

- Model hyperparameter tuning
- Experimentation with insurance specific synthetic datasets
- Visualisation of the overall output model
- Tool-assisted verification of constant-time properties
- Python framework/wrapper for more convenient usage