Privacy-Preserving Route Matching

Simplifications and additions

Master's thesis in Software Engineering

Gunnar Örn Gunnarsson

Pedram Talebi

Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG
Gothenburg, Sweden 2018

Master's thesis 2018

Privacy-Preserving Route Matching

Simplifications and additions

Gunnar Örn Gunnarsson
Pedram Talebi

Department of Software Engineering
Chalmers University of Technology
Gothenburg, Sweden 2018

Privacy-Preserving Route Matching
Simplifications and additions
Gunnar Örn Gunnarsson
Pedram Talebi

Supervisor: Andrei Sabelfeld, Department of Computer Science and Engineering
Advisor: Johan Josefsson, ICT i Göteborg AB
Examiner: Gerardo Schneider, Department of Computer Science and Engineering

Master's Thesis 2018
Department of Software Engineering
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: "Matching driver and passenger" - Image from BlancRide's webite, https://blancride.com/. The image represents the communication between two users of a mobile application.

Typeset in LaTeX
Gothenburg, Sweden 2018

Privacy-Preserving Route Matching
Simplifications and additions
Gunnar Örn Gunnarsson
Pedram Talebi
Department of Software Engineering
Chalmers University of Technology

Abstract

Due to recent security breaches in today's common applications, privacy requirements for modern applications have become stricter. To fulfill these requirements, more and more research has been put into producing methods that preserve the privacy of users while retaining the same degree of functionality and efficiency.

In this thesis, based on the privacy-preserving ridesharing model PrivatePool, two novel models are presented for preserving user's privacy in a ridesharing application. The former model optimizes the algorithm of PrivatePool to increase its efficiency, while the latter model takes additional parameters into consideration when performing its computations. Finally, we conclude that an ad-hoc privacy-preserving algorithm outperforms a general solution and that the addition of more parameters to the ad-hoc model has a significant negative effect on its efficiency.

Keywords: ridesharing, privacy, optimization, security, matching, secure mulity-party computation, private set intersection, endpoint, intersection, PrivatePool.

Acknowledgements

Firstly, we would like to express our sincere gratitude to our supervisor, Andrei Sabelfeld, for introducing us to this subject, guiding us through our work and believing in all our ambitious goals. We would like to thank Gerardo Schneider for being our examiner and for being so helpful throughout the whole process. From the bottom of our hearts, we would like to thank Johan Josefsson, the whole Ictech family, Per Hallgren, our significant others, parents, siblings and, especially, everyone that dedicated their time and efforts into reading the report and giving invaluable feedback.

Gunnar Örn Gunnarsson & Pedram Talebi, Gothenburg, June 2018

Contents

List of Figures

List of Tables

1

Introduction

In recent years, security requirements for modern applications have increased significantly, due to the amount of sensitive data which they process. What can be categorized as sensitive data, depends on the context. However, the term generally refers to data that is not publicly available and can be associated to a specific person or a group of people. Applications are used on a daily basis, that often process and handle private information. Credit card numbers, locations of users and/or social security numbers are some examples, that are required by applications to fulfill their purposes. However, this also increases the risk of users' data being abused or falling into the wrong hands, which could reveal more information about the user than the user has explicitly agreed to. Despite the fact that third party applications should handle their customers' data with care, this is not always the case. It turns out that customers' data could be used for privacy-violating activities. Up until now, there have been several cases of service providers abusing their customers' data in such a manner [16, 28, 43].

Ridesharing applications are examples of privacy-sensitive applications. The intuition behind ridesharing is that two or more parties are willing to share a ride, given the condition that they intend to travel similar routes or between similar endpoints. Benefits of ridesharing include the ability to commute without having access to a personal motorized vehicle and reduction of travel cost and the environmental footprint. Uber [20], Lyft [19] and Blablacar [2] are examples of ridesharing applications, with Uber being the most prominent application in the western markets [53].

Several methods are available to enhance the security in privacy-sensitive applications [5, 3, 15, 14, 17, 38, 26, 37, 52]. In this thesis, improvements are made to a decentralized and distributed model that provides a strict security protocol for a ridesharing application.

1.1 Background

Abuse cases in ridesharing applications are in part what motivated Hallgren et al. to develop PrivatePool [15], which is a model for privacy-preserving ridesharing. The goal of PrivatePool is to reveal to its users a commonly shared part of the routes they intend to traverse, only if the routes are similar enough and fulfill the user's requirements, and no information otherwise. By utilizing *secure multi-party computations* (SMC) [51], which is a subfield of cryptography, participants can jointly compute a function based on private inputs. SMC protocols can be used to enable users to detect whether other users have similar routes, in order to share a ride with them, without revealing their positional information. In order to determine the degree of similarity between two routes, PrivatePool utilizes a variant of a *private set intersection* (PSI) protocol [17]. A PSI protocol computes the intersection between two sets, which makes it possible to determine whether they include two identical data points. The variant of PSI used by PrivatePool is called *threshold private set intersection* (T-PSI), which makes it possible to determine whether the two routes overlap above a given threshold.

The implementation of a real case usage of the PrivatePool model requires the ability to scale for many users and also to match them based on the time of their ride. In a scalable application with many end-users, the security aspects of the application is of utmost importance. Therefore, the privacy of each user is critical, and a dependable and privacy-preserving authentication method should be investigated. The time and location of a ride are essential parameters that are required to detect a possible rideshare. However, the time of the ride was not included in the PrivatePool model. The exclusion of the parameter results in a model with limited capabilities, which needed to be addressed for further progress towards a production-ready application. Additionally, the current model makes the assumption that all parties are aware of each other's existence and that they can trust that other parties are authenticated users.

The most significant drawback of PrivatePool is the solution's worst-case computational overhead, indicated in running time. Running time is defined as the time it takes for an application or a part of an application to fulfill its predetermined tasks. In this thesis, two models are proposed called *Optimized PrivatePool* (O-PrivatePool) and *Time-extended PrivatePool* (T-PrivatePool). This thesis is a continuation of and based on PrivatePool [15], where the aforementioned limitations are addressed. The scope includes optimizing the worst-case running time, adding time as a matching parameter and providing suggestions for authentication methods.

Since PrivatePool is designed in the context of ridesharing, it is logical to continue the research in the same context. By comparing PrivatePool with O-PrivatePool and T-PrivatePool under identical conditions, a reliable and fair comparison can be made. The objective is to provide a solution that can be generalized and applied to any number of contexts where the goal is to group entities together that share similar routes.

1.2 Purpose and Objectives

The purpose of this thesis is to further research and develop two new models, O-PrivatePool and T-PrivatePool, which address the limitations of its predecessor, PrivatePool. The main focal point is to investigate whether the alternative specialized solution O-PrivatePool based on a PSI protocol, presented in Chapter 4, can outperform the T-PSI based solution which utilizes a threshold key encapsulation mechanism (T-KEM) presented in PrivatePool [15]. The optimization is expected to reduce the computational overhead and decrease the worst-case time complexity of the model's so-called intersection-based matching method (from $\mathcal{O}(n^3)$ to $\mathcal{O}(n)$). More precisely, the running time of the method is expected to be significantly less than the former. Furthermore, it is expected that the model will suffer no loss in effectiveness, where effectiveness is the degree to which the model can detect all available ridesharing opportunities.

Based on the aforementioned statements, the following research questions are presented:

RQ1: *Will the new proposed O-PrivatePool's intersection-based matching method have less time complexity (denoted by T(n)) than PrivatePool's corresponding method based on T-KEM, which is bounded by $\mathcal{O}(n^3)$? I.e.:*

$$T(n) < \mathcal{O}(n^3)$$

RQ2: *Will the implementation of O-PrivatePool's intersection-based matching method result in a significantly lower running time, when running benchmarking tests, compared to PrivatePool's corresponding method?*

Requesting a rideshare requires the matching of trajectories and the time the ride is expected to take place. Introducing a parameter that represents time into O-PrivatePool is addressed further in Chapter 5. A potential solution, called T-PrivatePool, involves the user's routes being represented as routes in both space and time. Currently, the parties have a threshold parameter for how much they are willing to deviate from their route, in terms of spatial distance. A similar temporal deviance parameter would also be taken into consideration when redesigning the protocol. Therefore, the following research question is proposed:

RQ3: *Is it possible to match entities using an additional threshold, representing deviation in time, without significantly impairing the protocol's efficiency?*

In PrivatePool, it is assumed that all users utilizing the service are authorized and verified. However, the reality is different; data breaches [27] and unauthorized accesses [22] are well known issues of these types of applications. Since PrivatePool is intended to be utilized in a decentralized and distributed system, there is the potential of applying a decentralized and distributed authentication service to address some of the limitations of the protocol [32]. In Chapter 6, authentication services and their cryptographic principles are discussed at a high level and how they can be applied to the model.

1.3 Delimitations

In regards to SMC protocols, other potential alternatives than the new proposed protocols are not investigated, including the possibility of extending the former T-KEM protocol. This is due to the fact that other protocols have not been reported to provide the same degree of improvements.

Since all research surrounding the authentication aspect is open ended, only a high level description regarding its applications in this context is presented. The description is limited to the possibility of authenticating the user in a decentralized and/or distributed system. An implementation of the recommended authentication method is not presented in this thesis.

1.4 Contribution

The novel contributions of this thesis are two privacy-preserving ridesharing models, O-PrivatePool and T-PrivatePool. Both models are optimized versions of PrivatePool, while T-PrivatePool includes additional parameters for matching entities according to the time of their ride and the users' willingness to deviate from that time.

The theoretical evaluation of the algorithm given in Section 4.1.1 shows that O-PrivatePool achieves a significant decrease in worst-case time complexity, from $\mathcal{O}(n^3)$ to $\mathcal{O}(n)$. Furthermore, our practical evaluation of the computational overhead measurements, given in Appendix A and analyzed Section 4.4, showed that O-PrivatePool can produce significantly lower computational overhead than its predecessor, while retaining the exact same degree of effectiveness, see Section 4.4.1.1. Furthermore, theoretical evaluations of the algorithmic solutions of T-PrivatePool, provided in Section 5.1.1, showed that the worst-case complexity of one of the model's main functions is increased from $\mathcal{O}(n)$ to $\mathcal{O}(n \times m)$. Additionally, practical evaluations of measurements of the computational overhead, given in Appendices B through D and analyzed in Section 5.2, show that the model is indeed impaired to a statistically significant extent.

1.5 Outline

In Chapter 2, fundamental concepts and work in the related field are presented, such as PrivatePool. Chapter 3 discusses the research method used throughout the project and its phases, in addition to the terminology used for statistical hypothesis testing and analyses. The procedure performed whilst developing the first novel model, O-PrivatePool, is presented in Chapter 4, as well as the results and analyses of its efficiency and effectiveness tests. In Chapter 5, a similar process as in its preceding chapter is conducted, in order to develop and evaluate T-PrivatePool, by applying analogous experimental methods and analysis methods. Chapter 6 provides the reader with a high-level description of authentication methods and recommendations for a method that could potentially be applied to PrivatePool or its variants. In Chapter 7, the hypotheses presented for both O-PrivatePool and T-PrivatePool are evaluated. Finally, in Chapter 8, the novel contributions are evaluated in a broader context and potential future work is presented.

2

Theoretical Background and Related Work

2.1 Secure Multi-Party Computation

SMC is a subfield of cryptography which has the aim of enabling a set of parties $p = p_1, \ldots, p_n$ with private inputs $x_1 \ldots x_n$, to compute a public joint function $f(x_1, \ldots, x_n)$ on their inputs while retaining their private characteristics [51]. By utilizing SMC protocols, the reliance on an outside party to keep a secret is minimized or eliminated. Parties exchange messages between each other and can only interpret the output and their own input.

Despite the fact that SMC protocols have existed since the early 1970's, the protocols have been considered impractical in production level applications. However, according to Claudio Orlandi [35], applications using ad-hoc SMC solutions are constantly moving towards what can be considered realistic production level, in terms of efficiency.

2.2 Private Set Intersection

PSI is a branch of SMC which allows two parties, S and S', to compute the intersection of their private inputs $I = S \bigcap S'$, while revealing no information other than the intersection itself. Several different types of PSI protocols have been proposed over the years, categorized as generic or customized for the specific application. Generic PSI protocols work over garbled circuits (a cryptographic protocol, described as a Boolean circuit) [52, 38, 37] to compute the intersection, meanwhile, custom PSI protocols work over homomorphic encryption and other public-key techniques [5, 3, 26]. The state-of-the-art protocols which have shown the most advancements are custom protocols. However, recent research has been made on generic PSI protocols by Pinkas et al., concerning a new Efficient Circuit-based PSI via Cuckoo Hashing, whose efficiency is concretely better than other existing constructions [38]. As previously mentioned, custom protocols have made great progress, where *batched oblivious prf* (BaRK-OPRF) is one of the fastest state-of-the-art custom PSI protocols [26].

2.2.1 Threshold Private Set Intersection

A standard PSI protocol reveals the intersection as soon there is a match. T-PSI, on the other hand, is a variant of PSI which determines whether the two users' sets overlap, over a given threshold t [15]. If that requirement is fulfilled, the overlapping subset is revealed, otherwise nothing is revealed. In the context of ridesharing, T-PSI can be used to detect whether two routes overlap and if the length of the overlap exceeds the threshold t. In that case, both users are notified that they have the possibility of sharing a ride for that segment of the route.

2.2.2 Threshold Key Encapsulation Mechanism

T-KEM is a cryptographic tool which was used in PrivatePool [15] to construct a T-PSI protocol. T-KEM only reveals the intersection of two datasets if the number of intersecting elements exceeds a given threshold. The protocol is based on Shamir's secret sharing scheme [41], where participants are only able to generate a key if a certain number of valid points along a specific path are provided. The key which is released by T-KEM is then used to decrypt the intersecting values. Shamir's secret sharing has a complexity of $\mathcal{O}(n^3)$ which T-KEM inherits and is considered to be the most significant drawback of the tool. It should be emphasized that T-KEM can be applied to a wide range of contexts.

2.3 Symmetric and Asymmetric Encryption

In cryptography there are two variants of cryptosystems: symmetric and asymmetric [49]. In the former, both encryption and decryption of the plaintext/ciphertext is done by utilizing the same cryptographic key. For asymmetric systems, two different keys are generated, one public and one private. The public key, in that case, is publicly available and is used for encrypting a message, meanwhile the private key is only held by its intended recipient and is used for decrypting the message.

2.4 Homomorphic Encryption

Homomorphic encryption is a form of encryption that allows the ciphertext to retain the plaintext's homomorphic properties [5]. Additionally, linear operations performed on the ciphertext are reflected on its corresponding plaintext. After applying linear operations, e.g. additions and multiplications, to the ciphertext, it can be decrypted in order to retrieve the plaintext with the applied operations [7].

There exist cryptosystems with fully and partially homomorphic encryption. Fully homomorphic protocols allow unlimited amount of linear operations to be performed, while partially homomorphic protocols limits the number of operations. However, for practical use, the former is considered too computationally demanding. Therefore, majority of today's cryptosystems with homomorphic properties are based on partially homomorphic encryption. Paillier [36] and ElGamal [6] are two well known cryptosystems.

2.5 Proximity Testing

It is common that applications use the location of their users for either supporting core features or to enhance the user experience. A common case, which is of particular interest for this research, is when the user checks whether their associates are nearby [14].

Proximity Testing in a privacy-preserving environment involves computing whether users fulfill the proximity requirement without revealing their distance and relative positions to each other or to any third party. There are several privacy-preserving solutions for proximity testing with different strengths and weaknesses [42, 9, 31, 10, 54, 29, 47, 39, 14].

2.6 PrivatePool

PrivatePool is a novel model for privacy-preserving ridesharing [15]. The goal of the model is to provide a fully functional ridesharing application without compromising the privacy of its users. Coordinates are encrypted and SMC is applied over the encrypted inputs to detect ridesharing opportunities. Whenever two users fulfill the following requirements, a ridesharing opportunity emerges: Both users are travelling similar routes or between similar endpoints and they are willing to share a ride together. From the perspective of detecting whether the first requirement of a ridesharing opportunity is fulfilled, there are two patterns that are taken into consideration, *endpoint-based matching* and *intersection-based matching.*

In PrivatePool, the method of generating routes (or trajectories) while preserving privacy are considered out of scope. However, an assumption is made that the routing software computes the shortest path between the user's origin and destination. In Section 1.1, attention is drawn to the drawbacks of PrivatePool. One of those drawbacks is the fact that the time of the ride is not included in the model. Therefore, when referring to the shortest path between two points, it indicates its physical distance.

2.6.1 Ridesharing Patterns

A ridesharing pattern, called the *endpoint-based matching* pattern, has the goal of matching two parties' respective origins and destinations, without revealing their positions and distance to each other or any third party. In order to calculate whether the two participants' distance is within predetermined limits, a proximity test is made with homomorphic encryption.

Another pattern in ridesharing, the *intersection-based matching* pattern, has the goal of matching two parties only if there exists a sufficient overlap of their routes, without revealing any sensitive information while computations are carried out. A variant of PSI is applied to solve the intersection matching, since a regular PSI protocol reveals the intersection once there is a single match. Given that it is desired that two individuals share at least a minimum threshold length of the route, applying this protocol would not be optimal in a ridesharing context. It was, therefore, determined to construct and utilize T-KEM for their solution.

Fig. 2.1 and 2.2 present examples of how ridesharing opportunities may occur by applying endpoint-based matching and intersection-based matching. Fig. 2.1 depicts the scenario when both start- and endpoints are close to each other and the routes do not intersect. In this case, the two parties might consider sharing their trip instead of traversing two individual routes.

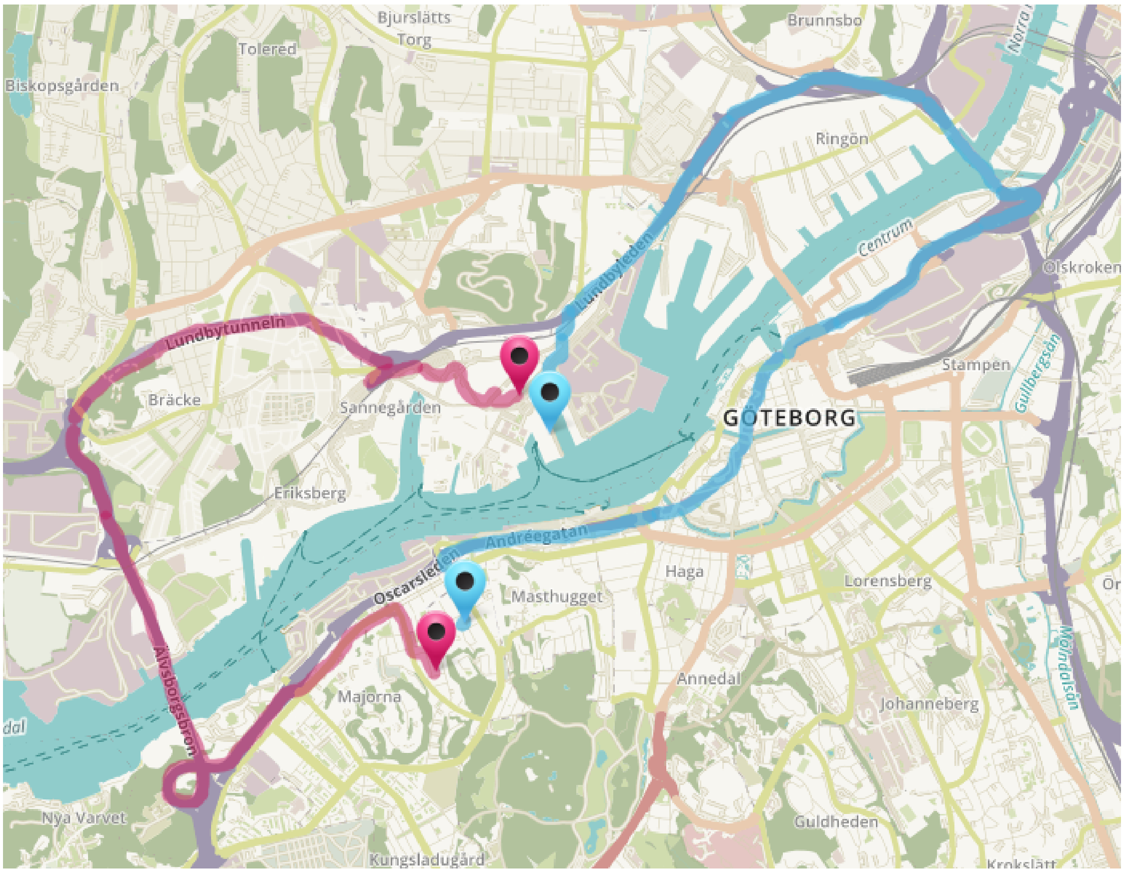

Figure 2.1: Two routes that have endpoints close to each other but do not intersect

Fig. 2.2 on the other hand, shows the scenario when the endpoints are not close to each other. However, it can be shown that both users' routes overlap to a certain extent. This overlap makes up a large part of the trip which makes it appropriate to share a ride.

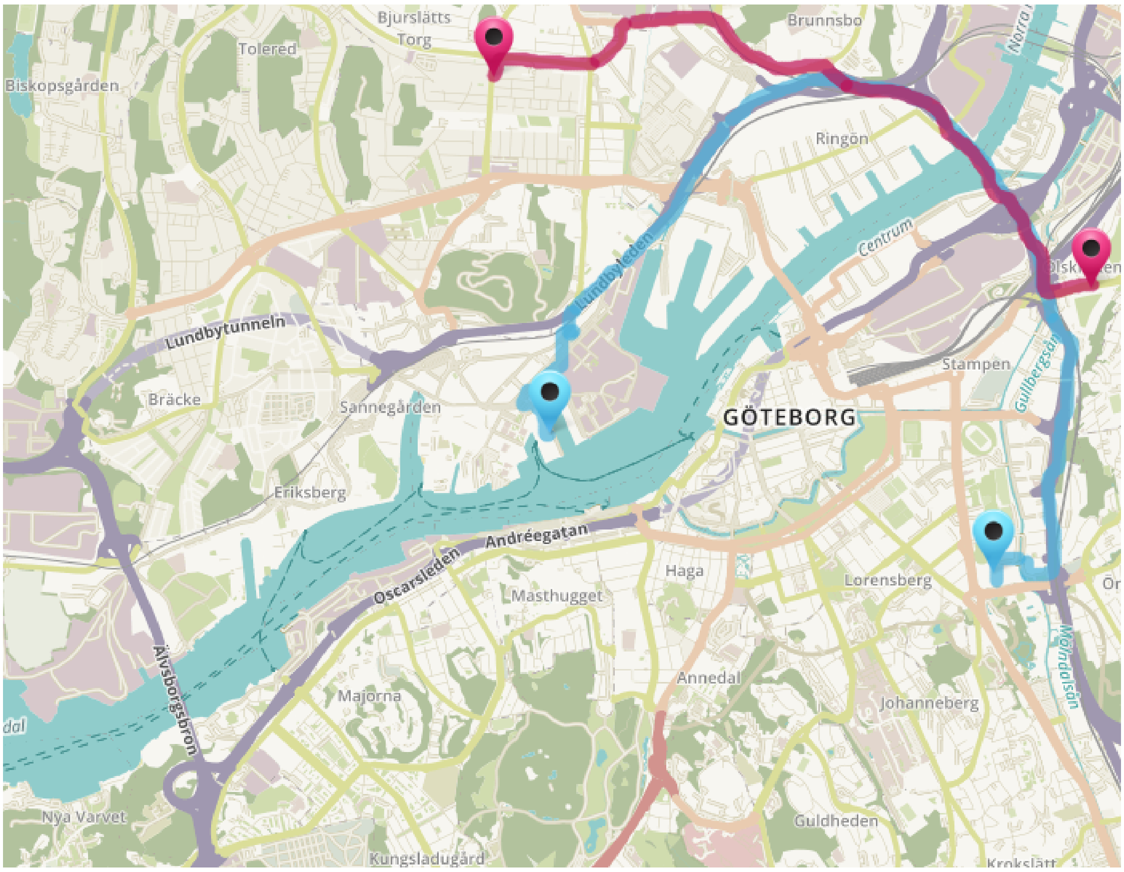

Figure 2.2: Two routes that intersect but their endpoints are far from each other

2.6.2 Ridesharing Modeling and Feasibility

In order to gain basic knowledge of the terminologies and concepts used in the ridesharing model, a number of definitions extracted from PrivatePool [15] are presented. In addition, all definitions are based on the fact that users travel with constant speed and that both the spatial and temporal cost of traversing a road section is equivalent.

Trips are represented as undirected, unweighted graph $G = (V, E)$, where V is the set of vertices and $E \subseteq V \times V$ is the set of edges. Other synonyms for trips are routes, paths and trajectories.

Definition 1 (Trip) *Given a graph $G = (V, E)$, a trip T is an acyclic sequence of consecutive vertices $v_i \in V$, where $v_s = v_0$ is the origin and $v_f = v_{|T|-1}$ is the destination, such that $(v_i, v_{i+1}) \in E$ for all $i \in \{0, ..., |T| - 2\}$.*

Definition 2 (Segment) *Given a graph $G = (V, E)$, a segment S of a trip T in G is an acyclic sequence of consecutive vertices $v \in V$, such that S is a subsequence of T.*

Definition 3 (Segment length) *Given a segment S for some trip through a graph $G = (V, E)$, let $l(S) = \sum_{i=0}^{|S|-2} d(S[i], S[i+1])$, where $S[j]$ is the jth vertex in S and $d_x(p_1, p_2)$ is the Euclidean distance between the two points p_1 and p_2.*

A rideshare is considered feasible when there is a low enough cost and high enough benefit for both parties. The feasibility of a rideshare can also be referred to as a ridesharing opportunity and is modeled by two parameters. One parameter stands for the upper limit of how much the user is willing to deviate from their trajectory, before and after their shared ride. The second parameter stands for a lower limit, or threshold, of the length of the ridesharing segment. A formal representation of ridesharing feasibility is given in Definition 4. Note that the deviation function $\mathit{\Delta}$ in Definition 4 can be an arbitrary function, that represents the change in how much an entity is willing to deviate from their route at a specific point.

Definition 4 (Ridesharing feasibility) *For any fixed threshold t and deviation function Δ, given two trajectories P_A and P_B for users A and B in $G = (V, E)$, ridesharing is feasible for A along a segment $S = \{p_s, ..., p_f\}$ of P_B if and only if $l(S) > t$ and there exist two points $P_A[i]$ and $P_A[j]$, with $i < j$, such that:*

$$d(P_A[i], p_s) < \Delta_{P_A}(P_A[i])$$

$$\wedge\ d(P_A[j], p_f) < \Delta_{P_A}(P_A[j]).$$

3
Research Method

Several research methods are available when performing quantitative research in the realm of Computer Science and Software Engineering [46, 24, 30, 50, 12]. *Design science research* (DSR) in Information Systems is a research method that captures the purpose of this thesis [46]. DSR involves two primary activities, where the first is comprised of the creation of a novel or innovative artifact and the second is an analysis of the artifact's use and performance. The artifacts involved are the new models O-PrivatePool and T-PrivatePool, and the analysis involves measuring the efficiency of each protocol compared to its relative change in effectiveness and then to analyze the results in different contexts.

A DSR is of iterative nature, where each iteration is made up of five phases: *Awareness of problem*, *Suggestion*, *Development*, *Evaluation* and *Conclusion*. These phases are iterated upon as many times as necessary. In this thesis, there are three iterations in total, one iteration is reserved for Chapter 4 and then two iterations are conducted in parallel, in Chapter 5 and Chapter 6.

In Section 3.1, the phases of DSR are explored in greater detail. Section 3.2 is presented to help the reader understand the basic terms used in the subsequent parts of the thesis. Advanced descriptions of methods used in statistical hypothesis testing are considered out of scope.

3.1 Design Science Research

The *Awareness of problem* phase was considered a precursor to this thesis, where a literature review was carried out. In general, the literature review involved investigating recent developments in the field. That is, the state of today's privacy-preserving applications, familiarizing with concepts in the ridesharing domain and researching the potential of applying decentralized and distributed technologies for the ridesharing scenario. Literature was discovered by systematically reading through PrivatePool's references and by utilizing Google Scholar [11] to search for papers that have referenced PrivatePool. In addition, Google Scholar was used to search for the most recent developments in the field by using keywords such as SMC, PSI, T-PSI, Homomorphic encryption, Proximity Testing and other variations of those keywords. For each iteration, the relevant knowledge acquired from the *Awareness of problem* phase was used to produce the design during the *Suggestion* phase.

During the *Suggestion* phase, from the motivation of the relevant research questions, requirements were elicited and outlined. Subsequently, hypotheses were proposed from their corresponding research questions, along with a detailed design for a suggested solution and an experimental design.

Physical and virtual machines with the required software were set up for the *Development* phase and subsequent simulations. By utilizing the detailed designs generated from the *Suggestion* phase, an implementation was accomplished. That included the development of necessary supplementary programs, such as test suites. In addition, benchmarks were made on a dedicated machine with the following system information:

Manufacturer: *Dell Inc.*
Product Name: *PowerEdge T20*
Specifications: *16GB RAM and an Intel(R) Xeon(R) CPU E3-1225 v3 @ 3.20GHz.*

During the *Evaluation* phase, tests were designed in accordance with the experimental design and simulation data was gathered and prepared for testing. Evaluations of the new models, O-PrivatePool and T-PrivatePool, were carried out using the same methods. The evaluation methods included a worst-case algorithmic complexity analysis, a statistical analysis on the running times of the intersection-based matching method and an effectiveness test. T-PrivatePool was evaluated with a supplementary statistical analysis to determine whether an additional parameter, representing deviation in time, has a significant effect on the endpoint-based matching method. Measurements of the model's running times were made by utilizing the same real-world data as in PrivatePool. The real-world data was publicly provided by the New York City Taxi & Limousine Commission (TLC) on their website and were believed to represent realistic activity of ridesharing in a densely populated area.

During the *Conclusion* phase, the statistical analyses that were produced during the preceding *Evaluation* phase were put into context and used to evaluate the stated hypotheses.

3.2 Terms for Statistical Hypothesis Testing

Statistical hypothesis testing, or simply hypothesis testing, was used as the method of choice to answer the given research questions in Section 1.2. The purpose of this section is to inform the reader about the fundamental terminology required, in order to follow the reasoning behind hypothesis testing. All terms and procedures described are derived from [23].

3.2.1 Parameters and Variables

While performing experiments, data is gathered by observing the *response variable*. An *experimental unit* is the subject of the experiment. In addition to the response variable, controlled independent variables are usually included in the experiment, monitored and manipulated by the experimenter. These variables are referred to as *factors* and are used to determine whether they have an effect on the response variable. Factors assume pre-defined and limited number of possible values, which are known as *factor levels*. Each factor has one or more levels and they can be seen as number of variations of the factor that is used in the experiment. In addition, combinations of factor levels are called *treatments*.

3.2.2 Statistical Hypotheses

Statistical hypotheses, or simply *hypotheses*, are statements that can be either rejected or failed to be rejected, by applying hypothesis testing. Hypotheses are often formed from the specific goal, aim or purpose of the study being undertaken. Formally, a hypothesis is a statement about whether a specific treatment has an effect on a measured value gathered from the sampled population. The population is often assumed to be infinite or very large and is, therefore, sampled. Thus, the mean (μ) of the gathered values from different samples are compared in order to answer the hypothesis. In this thesis, for example, hypotheses can be formed from the research questions given in Section 1.2.

Generally, hypotheses come in pairs, where one is called the null hypothesis and the second one is called the alternative hypothesis. A null hypothesis is typically the statement that the different factors, influencing the experimental unit, do not have an effect on the response variable. This is denoted by:

$$H_0 : \mu_A = \mu_B.$$

An alternative hypothesis is the counterargument to the null hypothesis. If the intention is to measure whether there is a significant difference in the treatments, no matter if the difference is negative or positive, then the alternative hypothesis is denoted by:

$$H_A : \mu_A \neq \mu_B.$$

This is referred to as a two-tailed test. If the researchers have an indication that one mean is either generally lower or higher than the other, a one-tailed test could be applied. The alternative hypothesis is then denoted by:

$$H_A : \mu_A < \mu_B.$$

3.2.3 Statistical Hypothesis Testing

Hypothesis testing is a statistical analysis method that applies an experimental process in order to evaluate a given hypothesis. The experimental process takes quantitative data into consideration when determining whether there is a statistically significant relationship between the means of two or more randomly sampled populations. In most cases, the populations have been subjected to two different treatments. In that case, if there is a significant difference, the treatment methods can be said to have a statistically significant effect on the measured outcome.

When performing a hypothesis test, statistical significance is denoted by an alpha level (α), where α indicates the maximum likelihood of there being a type I error (i.e. rejecting a true null hypothesis). In standard practice, α is most commonly set to 0.05 or 0.01. If α is set to 0.05 and a null hypothesis has been rejected, it can be stated that it has been rejected with a confidence of 95%. Note that using a one-tailed test means that the significance level α is decreased to be half of what its corresponding two-tailed test would have. The rationale behind this behaviour is beyond the scope of this thesis.

In the context of this thesis, hypothesis testing can be used to determine whether the time it takes the application to compute a ridesharing opportunity is affected by seemingly unrelated factors. These factors could include how far the user is willing to deviate from their route, how willing they are to deviate from their estimated time of departure or arrival and what the user require as the minimum length of their shared route.

3.2.4 Experimental Design

The outline, or arrangement, of an experiment is called an *experimental design*. Experimental design includes procedural instructions on what variables should be measured and what statistical information should be gathered. Additionally, it defines what treatments should be applied during each experiment, how many experiments should be run and how many times the experiments should be repeated. The experimental design of choice can vary depending on the purpose of the experiment, the number of factors that could influence the experiment, the number of levels of factors and whether some factors bring an unwanted source of variation into the response variable. Examples of design choices include *One-Factor designs*, *Block designs*, *Nested designs* and *Factorial designs*.

A One-Factor design is used when there exists a single factor of desired variation. The different levels of the factor are then applied to the experimental unit in sequence. A factorial design, however, is used when there exist multiple factors of desired variation. The effect of each factor is then quantified either by measuring each factor independently or by, additionally, measuring the interaction effects of multiple factors.

In some cases, due to time or budget constraints, researchers can opt to a fractional variant of their design. In a *fractional design*, a limited number of factors and/or levels of factors are investigated. The levels of factors that are used are carefully selected as representative values for those factors, while the factors that are chosen are either interaction effects or considered particularly interesting for the research.

4

Matching Operation

This chapter aims to guide the reader through the methods used during the development of O-PrivatePool. As mentioned in Section 1.2, the purpose was to provide a model which addresses the limitations of its predecessor, PrivatePool. During this iteration, a step was taken towards implementing a production-ready application, which does not suffer from significantly high computational overhead, or running times. The methods used during the iteration were carried out as outlined in Section 3.1. Section 4.1 lays the foundation for an experiment, Section 4.2 describes the development procedure, Section 4.3 encompasses the evaluation procedure. The results can be found in Section 4.4 and Section 4.5 explains the threats to the validity of the experiment and their mitigation strategies.

4.1 Suggestion

As mentioned in Chapter 1.2, the main focal point of this thesis was to investigate whether an alternative specialized PSI solution is more efficient than T-KEM, utilized in PrivatePool. Two indicators of an efficient algorithm that are taken into consideration include whether it has a polynomial solution and whether the running times are efficient. According to Kleinberg and Tardos [25], algorithms that have have a polynomial solution are considered as the most efficient and prominent ones. However, in a realistic setting, an algorithm's efficiency can be more precisely determined by measuring its running time. PrivatePool, which utilizes T-KEM, has a polynomial solution whose exponent is considered relatively high, given as $\mathcal{O}(n^3)$. Here, n is the number of segments that make up the total route. This high degree in algorithmic complexity is reflected in the application's running times, which consequently increase according to a cubed function as the input sizes increase linearly. To scale up and use this model in a realistic application, the exponent of the polynomial needs to be reduced. That means the new proposed O-PrivatePool solution has to have an exponent which is significantly lower. This draws the attention back to the first two research questions in Chapter 1.2:

RQ1: *Will the new proposed O-PrivatePool's intersection-based matching method have less time complexity (denoted by T(n)) than PrivatePool's corresponding method based on T-KEM, which is bounded by $\mathcal{O}(n^3)$? I.e.:*

$$T(n) < \mathcal{O}(n^3)$$

RQ2: *Will the implementation of O-PrivatePool's intersection-based matching method result in a significantly lower running time, when running benchmarking tests, compared to PrivatePool's corresponding method?*

The first research question, *RQ1*, can be answered by means of a worst-case time complexity analysis, following a detailed solution design. Therefore, it is not deemed necessary to perform a hypothesis test in this case. However, *RQ2* indicates the need to determine whether the new O-PrivatePool model provides a statistically significantly lower running time than PrivatePool. Here, the experimental unit is the ridesharing model and the response variable is the computational running time. Therefore, the following hypotheses are presented:

H_{01} : The running times of both intersection-based matching methods are the same,

$$\mu_{O-PP} = \mu_{PP}.$$

H_{A1} : The running time of O-PrivatePool's intersection-based matching method is significantly lower than PrivatePool's method,

$$\mu_{O-PP} < \mu_{PP}.$$

An additional requirement is that there is no loss in effectiveness when applying the two methods.

4.1.1 Proposed Solution

As discussed in Section 1.1, PrivatePool makes the assumption that the computed routes between two endpoints consist of the shortest distance between the two points. Therefore, it can be assumed that each two intermediate points between the endpoints are connected by their shortest distance. Thus, it is obvious that if a point i is present on both routes and the corresponding point $i + t$, at which the user would exceed the required intersection threshold, is also present on both routes, all intermediate points are guaranteed to be present in both routes. This indicates that it is only necessary to check whether a pair $(i, i+t)$ exists on both routes. The steps of the algorithm are presented in the *Future work* chapter in PrivatePool [13] (p. 203):

"[...]The simple scheme outlined in the following could outperform our solution based on T-KEM, where Alice has trajectory T^A size m and Bob a trajectory T^B size n, for any given threshold t.

Note that this solution only works when the input data is sorted as in our case for ridesharing. We need that if $\exists i, j : T_i^A = T_j^B \wedge T_{i+t}^A = T_{j+t}^B$ then it also holds that $\forall u \in \{i, ..., i+t\}, v \in \{j, ..., j+t\} : T_u^A = T_v^B . \forall u \in \{0, ..., t\} : T_{i+u}^A = T_{j+u}^B$.
The construction proceeds as follows:

1. Alice prepares a set on the following form,

$$(T_1^A, T_t^A), (T_2^A, T_{t+1}^A), ..., (T_{m-t+1}^A, T_m^A)$$

2. Bob similarly prepares a set on the form

$$(T_1^B, T_t^B), (T_2^B, T_{t+1)}^B), ..., (T_{n-t+1}^B, T_n^B)$$

3. The parties run a standard PSI protocol on these sets."

This method is what has previously been referred to as O-PrivatePool's intersection-based method.

4.1.2 Experimental Design

This section provides the layout of the experimental design for testing hypotheses H_{01} and H_{A1}, which involve measuring the change in efficiency when applying different set intersection protocols. Additionally, a description of the supplementary effectiveness test suite is provided.

4.1.2.1 Intersection-based Matching

Before it was determined which experimental design to apply, the variables involved had to be defined.

Based on hypothesis H_{A1}, it was decided to measure whether applying O-PrivatePool's intersection-based method would result in a significantly lower running time than applying PrivatePool's method. Thus, a one-tailed hypothesis test was applied. Using a one-tailed test, however, had the effect that α was reduced by half, from 0.05 to 0.025. This is due to the fact that the significance level α is most commonly set to 0.05 in scientific research.

The experimental unit and response variable, as previously mentioned, are the intersection-based method and computational running time, respectively. Given that the tasks of constructing the datasets and running the set intersection protocol (PSI or T-PSI) are completed in sequence, it was estimated that the running time is primarily bounded by the task which has higher algorithmic complexity. Therefore, the factors that were suspected to influence the response variable were the intersection based method that was applied and the length of the ride, where the length of the ride is indicated as the dataset size. The first factor in the experiment had two levels, a standard PSI protocol and T-KEM. The second factor had eight levels, i.e. the integer values 5 through 12. The levels of the dataset size factor indicates the exponent value n, which was used to compute the actual size 2^n.

Since there were two factors with several alternatives, it was evident that a Factorial design was appropriate. More precisely, given that the dataset size can take an infinite number of alternatives, a variant of a Factorial designs called Fractional Factorial design was applied. According to PrivatePool [15], the running times of T-KEM grow in a cubed function according to a linearly growing input size. Due to the magnitude of the estimated running times of T-KEM, the replications of the experiment were limited to ten replications per intersection-based method. A statistical regression tool was subsequently applied to analyze the measured data.

4.1.2.2 Effectiveness

Given that there was an additional requirement that there should not be any loss in effectiveness, a separate test was run in parallel to the efficiency tests. The effectiveness of the model was defined as the degree to which the model can detect all available ridesharing opportunities. The brute force algorithm, given by Definition 4, in Section 2.6.2, was believed to capture all possible ridesharing opportunities. However, during the literature review, there was no indication that there existed a privacy-preserving method that captured both endpoint-based and intersection-based patterns simultaneously. Therefore, the patterns were to be modeled individually and their efforts combined and compared to the total number of captured opportunities by the brute force algorithm. By comparing the number of detected ridesharing opportunities from both individual models to the corresponding brute force algorithm, it was possible to acquire the model's relative effectiveness.

Two primary variables that were used to model the ridesharing patterns were the maximum radius r and minimum threshold t. The maximum radius value r indicates the degree to which the user is willing to deviate from a specific point along their route, in terms of meters. The minimum threshold t denotes the minimum percentage of the user's total route, that needs to be exceeded in order to make the rideshare feasible.

4.2 Development

The goal of the development phase was twofold, firstly to collect measurements on the running times when applying the two models' intersection-based methods (i.e. the model's efficiency) and secondly to measure the difference in the models' ability to detect feasible ridesharing opportunities (i.e. the model's effectiveness).

The proprietary source code of PrivatePool and T-KEM were provided by the author of PrivatePool [15], along with their dependencies. The projects were initially set up on a virtual machine with limited memory and CPU specifications, for local trial experiments. The source code of PrivatePool came with a test suite for measuring the effectiveness of the model, while T-KEM also came with its own efficiency test suite for all the predetermined dataset sizes of the experimental design. Therefore, no development was needed for generating those tests. The effectiveness test for PrivatePool was reused as-is, in order to determine the effectiveness of O-PrivatePool.

Initially, the method of exporting the users' routes as datasets was altered. When considering a route or trajectory as a graph $G = (V, E)$, each edge E represents the individual paths of the routes and V represents the vertices or points along the route at which the edges meet. This can be written as the trajectory T_i^A, where A is a user and i is the ith point along the trajectory. Originally, PrivatePool exported the route's data in the following form:

$$(T_1^A, T_2^A, ..., T_m^A)$$

where m is the number of coordinates along A's trajectory. However, for the O-PrivatePool model, this exportation method was altered to export pairs of coordinates according to the solution given in Section 4.1:

$$(T_1^A, T_t^A), (T_2^A, T_{t+1}^A), ..., (T_{m-t+1}^A, T_m^A)$$

The first coordinate of every pair represents each individual node along the graph G and the second represents the coordinate of the node at which the segment would exceed the given threshold.

Note that PrivatePool was written in Python, while T-KEM was written in C++. Therefore, to minimize unfairness in both the models and the set intersection methods, it was determined to utilize the same programming languages for their respective replacements. As mentioned in Section 2.2, BaRK-OPRF is one of the fastest state-of-the-art custom PSI protocols. Therefore, BaRK-OPRF was selected as the PSI protocol for O-PrivatePool. Its source code was openly provided by the Cryptography research team at Oregon State University from their GitHub account [45]. BaRK-OPRF was then altered, using Microsoft Visual Studio [4], to read and parse the exported dataset and to return the result in an expected format. O-PrivatePool was then made to run the altered version of BaRK-OPRF. In order to evaluate the efficiency of the altered BaRK-OPRF protocol, an efficiency test suite, which was analogous and comparable to the one used to evaluate T-KEM and fulfills the requirements of the experimental design, was created.

In both efficiency tests, the protocols were made to perform the same tasks. In the first task, the protocols generated two datasets of equal sizes, that was predetermined by the test case. Half of each dataset was guaranteed to include identical elements, while the second half of each data set comprised of pseudo random elements. The second task was to mask the dataset with protocol-specific techniques, making it inexplicable to outside parties. The third task was role specific, where one party sent their data to the other and waited until the intersection had been computed, while the other party received the data and computed the intersection. The fourth task was to send the computed intersection to the waiting party. The final step was to present the computed data in a human-readable form.

4.3 Evaluation

In order to answer the first research question, a worst-case time complexity analysis of O-PrivatePool's intersection-based method was performed. Results generated by the time complexity analysis are displayed in Section 4.4. All tests suites, for PrivatePool and O-PrivatePool, were initially run on a local machine with limited specifications. The local runs were done in order to determine whether the test suites run as expected. The test suite for PrivatePool's efficiency was moved to the dedicated machine and executed ten times without interruption, according to the experimental design. Consequently, O-PrivatePool's efficiency tests were run under identical conditions on the dedicated machine. Results generated from the efficiency tests are provided in Appendix A and further analyzed in Section 4.4.1. PrivatePool's effectiveness tests were initially executed three times locally, from which it was evident that the results of the effectiveness test suite were not affected by the hardware specifications. Therefore, it was not deemed necessary to run the test suite more than once on the dedicated machine. Corresponding test suites for O-PrivatePool's effectiveness were executed under the same conditions. The results of the effectiveness tests are presented in Section 4.4.1.1.

As mentioned in Chapter 3, O-PrivatePool's effectiveness was evaluated on real-world data, publicly available from TLC's website [33]. TLC provides a large amount of data that can be used to generate endpoints of routes. However, the intermediate points of the routes are not included. Therefore, the open source routing software Routino [1], which uses open source mapping data from OpenStreetMap (OSM) [8], was used to generate intermediate points. Due to the significantly large amount of data provided by TLC, the data used for measuring effectiveness was limited to 1000 trips, from each even month of the year 2015. This same method of sampling the TLC data was used to evaluate PrivatePool's effectiveness. The r values of the endpoints, r_0, were set as 500 m, 1000 m and 2000 m, while the values of t were set as 20%, 50% and 80%. The r value of intermediate points, i along an entity's route were computed using the deviation function Δ given by Equation 4.1.

$$\Delta_T(i) = 4i^2\frac{r_0}{|T|^2} - 4i\frac{r_0}{|T|} + r_0 \tag{4.1}$$

Equation 4.1 is an equation extracted from PrivatePool. The equation gives a "happy-smiley-face" curve, where the endpoints' radius values reach r_0, while the radius values for intermediate points decrease as the index of the point approaches the median. The motivation for this function is that it is assumed that the user is willing to deviate the most from their route at the endpoints of their trips and not in the middle, due to the user being more familiar with the areas they come from or intend to go to.

After each experimental measurement, the outputs of each test were added to a Google Spreadsheet [18] document. Consequently, the regression tool called Statistics for Google Sheets (SGS) [44] was used to perform a regression analysis, using its built-in Regression feature. The results of the regression analysis and their evaluations are further presented in the subsequent chapter.

4.4 Results

The average running times of both T-KEM and the altered version of BaRK-OPRF are presented in Table 4.1, while the full table of measurements can be found in Appendix A. The first column of Table 4.1, *Dataset size* indicates the total number of coordinates (or segments) in both routes that are compared during each measurement. The supercolumn *Intersection-based method* indicates which method was applied during the measurements. The possible values of *Intersection-based method* can be seen by its subcolumns, *T-KEM* and *Altered BaRK-OPRF*. The role *Sender* indicates the instigator of the protocol, while the *Receiver* is the party which is queried and carries out the comparison itself. Therefore, the total running time, represented by *Total*, can be measured as the combined running time of both the computation on the *Sender*'s side and on the *Receiver*'s side. A graphical representation of the *Total* running times, from Table 4.1, is given in Fig. 4.1, where the vertical axis is represented as a logarithmic scale.

	Intersection-based method					
Dataset	T-KEM			Altered BaRK-OPRF		
size	Sender	Receiver	Total	Sender	Receiver	Total
32	0.0127	0.0203	0.0329	0.0821	0.0048	0.0869
64	0.0488	0.1531	0.2019	0.0823	0.0048	0.0871
128	0.1933	1.1973	1.3906	0.0817	0.005	0.0867
256	0.7732	9.5163	10.2894	0.0851	0.006	0.0911
512	3.0861	75.9819	79.0680	0.0831	0.0067	0.0898
1024	12.3264	606.3483	618.6747	0.0827	0.0086	0.0913
2048	49.2486	4843.9960	4893.2446	0.0835	0.012	0.0955
4096	196.5826	38630.7200	38827.3026	0.0839	0.0147	0.0986

Table 4.1: Average running times from ten replication per test, displayed in seconds

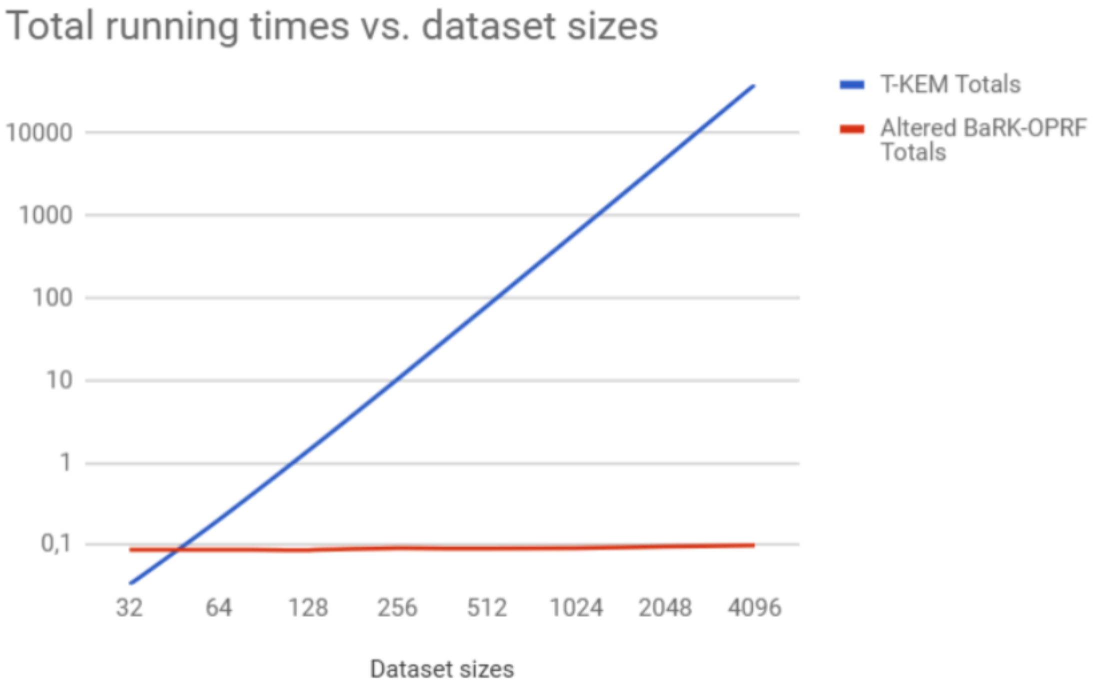

Figure 4.1: Graphical representation of total running times, using T-KEM (blue line) and the altered BaRK-OPRF protocol (red line) with varying input sizes

As seen in the proposed solution, outlined in Section 4.1.1, the construction of the two datasets in the first two steps can be carried out by looping linearly through the points at the first $n - t + 1$ indices and pairing them with the corresponding point at the indices t through $n+1$. The pairing activity consists of simple constant operations and, therefore, the worst-case complexity of the first two steps are given as

$$\mathcal{O}(n + n) = \mathcal{O}(n).$$

Furthermore, the two building blocks of the matching operation, the dataset generation and the PSI protocol run in sequence. Thus, the lower time bound of the operation is set at the data generator's linear running time while the upper bound is set by the complexity of the PSI protocol of choice. In this thesis, even though the practical evaluations might indicate that the PSI protocol has a constant running time, the theoretical complexity of the protocol might be of a different nature. However, PSI technology is constantly evolving and it is certainly possible that PSI protocols with linear, sublinear or constant running time might appear in the future. In the context of this evaluation, the complexity of the PSI protocol of choice can, therefore, be abstracted. In this regard, it can be stated that the intersection-based method's worst-case complexity is bounded by the $\mathcal{O}(n)$ complexity of the dataset generation method, while the choice of the PSI protocol could potentially influence the boundary.

4.4.1 Statistical Analysis of Efficiency

By using SGS' *Regression* feature on the data supplied in Appendix A, Tables 4.2 through 4.5 were generated. SGS attempts to generate a model that predicts the value of the response variable, based on the values of the given factors. A prediction model is generally described by Equation 4.2.

$$\hat{Y} = C_0 + C_A X_A + C_B X_B + C_{AB} X_A X_B + e, \tag{4.2}$$

where $\hat{Y}$ is the response variable, the C variables are the coefficients, X are the independent variables and e is the error, which the estimated value deviates from the measured value. In this case, SGS has produced the coefficient table, as displayed in Table 4.2. The first column of the table lists the factors that are included in the model. The second column, *Estimate*, provides the estimated value of each corresponding coefficient. The third column, *Std. Error*, gives the standard error of the coefficient, indicating the statistical uncertainty in a statistic. The regression tool takes the *Estimate* and *Std. Error* values into account when computing the *t value*, presented in the fourth column. The *t value* tells the analyst how many standard errors the estimated coefficient is from zero. If a coefficient is zero then the corresponding factor does not have an effect on the response variable. The value in the fifth column, *p value*, signifies the probability that a coefficient this large can be observed if the factor does, in reality, have no effect. A way to read the *p value* is to compare it to the α value of the experiment. If the *p value* is less then the value of α, the coefficient can be evaluated as non-zero and, therefore, has a significant effect on the response variable. Note that, for the coefficient $Intercept$, the *p value* is not above the given α level 0.025. However, the *p values* for the other coefficients fulfill that requirement.

Name of coefficient	**Estimate**	**Std. Error**	**t value**	**p value**
Intercept	1 036.345	870.437	1.191	0.2356
Dataset Size	4.429	0.408	10.856	<0.0001
Protocol (T-KEM = false)	-5 553.685	1 081.192	-5.137	<0.0001

Table 4.2: Coefficients and their corresponding standard error, t and p values generated by running SGS on the running times

Table 4.3 displays values that can be used to evaluate the overall goodness of the fit of the prediction model suggested by SGS. The *R-square* parameter is used to describe how much variation in the measured value can be explained by the model, which is in this case 47.9%. An adjustment can be made to the *R-square* by taking the degrees of freedom into consideration. This value is generally considered a better representation of how much of the variation can be explained by the model. The adjusted *R-square* value is represented by the *Adj R-square* variable, which in this case is 47.2%. *Residual SD* indicates the standard deviation of the residuals, where residuals are the differences between a measured value and the estimated value. *Sample SD* signifies the overall standard deviation of the measured values from the sample's mean. *N observed* and *N missing* represent the number of observed measurements and the number of missing measurements, respectively.

Variable	**Value**
R-square	0.0479
Adj R-square	0.472
Residual SD	6 838.056
Sample SD	9 411.991
N observed	160
N missing	0

Table 4.3: Parameters indicating the overall fit of the prediction model

The corresponding *ANOVA* table, generated by SGS, is presented as 4.4. An *ANOVA* table compares the variance in residuals to the variance of the overall data. Primarily, this table is used to determine whether the suggested prediction model explains the variation in the response variable to a significant extent. The first column represents the possible sources of variation, while the second column, *Df*, represents the degrees of freedom. One degree of freedom is said to be generated for each observed value and one degree is retracted for each estimated coefficient. The *Sum Sq* presents the sum of squares of the deviations of all observations from their mean. The *Mean Sq* indicates how much deviance can be explained per degree of freedom or, more generally, the column gives the reader an intuition of how well the model explains the variance. If the *Mean Sq* value is large, the model explains a considerable amount of the variation. The *F value* normalizes the *Mean Sq* value and helps the reader better evaluate the statistical significance of the model. If the *F* *value* is close to 1, it can not be determined whether the model does a useful amount of explaining. However, if the *F value* is larger than 1, it can be concluded that the model can explain a useful amount of the variance in the response variable. The *p value* in the *ANOVA* table is applied in the same manner as for the coefficient table; if it is lower than the given α level, the model can be said to explain a statistically significant amount of the variance in the response variable. Note that in this case, the *p value* is, indeed, lower than the α level 0.025.

ANOVA Table					
Factors	**Df**	**Sum Sq**	**Mean Sq**	**F value**	**p value**
Model	2	6.74×10^9	3.37×10^9	72.114	<0.0001
Residual	157	7.34×10^9	4.68×10^7		
Total	159	1.41×10^{10}	8.86×10^7		

Table 4.4: ANOVA table derived from the measured data in Appendix A

The final table generated by SGS, displayed in 4.5, provides *partial F tests* for each individual factor included in the prediction model. The columns of the table are analogous to the columns of the *ANOVA* table. *Partial F tests* are used to individually determine whether the factors can account for a significant amount of the variance in the response variable, within the model. Observe that the *p value* of both factors are lower than the α level of the experiment.

Partial F Tests					
Factors	**Df**	**Sum Sq**	**Mean Sq**	**F stat**	**p value**
Dataset Size	1	5.51×10^9	5.51×10^9	117.843	<0.0001
T-KEM = false	1	1.23×10^9	1.23×10^9	26.385	<0.0001

Table 4.5: Partial F tests for individual factors included in the prediction model

4.4.1.1 Analysis of Effectiveness

The results displayed in Tables 4.6 and 4.7, were produced from the protocols' effectiveness test suites. The values given represent the average percentage of matching opportunities detected by the ridesharing patterns, compared to a brute force approach which compares each individual point along both respective routes, while applying varying restriction parameters. Here, t represents the minimum threshold length of how much of the driver's route needs to be shared between both entities. However, the r_0 parameter, represents the maximum radius distance between the two endpoints. The radius value of intermediate points are, as described in Equation 4.1, dynamically computed using the value r_0 and the index of the point. Table 4.6 illustrates the effectiveness of PrivatePool while Table 4.7 represents the effectiveness of O-PrivatePool. A simple observation shows that both effectiveness tables are identical. The values in the columns *IS* and *EP* indicate the percentage of the effectiveness of the intersection-based matching and endpoint-based matching patterns, respectively. The columns *PP* and *O-PP* indicate the combined percentage values of the individual patterns.

t	$r_0 = 500$			$r_0 = 1000$			$r_0 = 2000$		
	PP	IS	EP	PP	IS	EP	PP	IS	EP
20%	98.29	97.99	0.30	65.09	63.7	1.39	38.22	32.27	5.95
50%	80.33	79.42	0.91	48.14	44.55	3.59	26.72	15.88	10.84
80%	31.31	28.8	2.51	16.07	9.96	6.11	15.38	2.48	12.89

Table 4.6: Average percentages of detected ridesharing opportunities in PrivatePool

t	$r_0 = 500$			$r_0 = 1000$			$r_0 = 2000$		
	O-PP	IS	EP	O-PP	IS	EP	O-PP	IS	EP
20%	98.29	97.99	0.30	65.09	63.7	1.39	38.22	32.27	5.95
50%	80.33	79.42	0.91	48.14	44.55	3.59	26.72	15.88	10.84
80%	31.31	28.8	2.51	16.07	9.96	6.11	15.38	2.48	12.89

Table 4.7: Average percentages of detected ridesharing opportunities in O-PrivatePool

4.5 Threats to Validity

In this section, potential unwanted aspects surrounding the experiment that might threaten its validity, are discussed. Here, validity threats are defined and evaluated according to Runeson and Höst [40]. The validity threats included in an experiment are used to denote the trustworthiness of the results that are presented in Section 4.4. Additionally, they can be kept in consideration when replicating the experiment. The validity threats are further discussed in Chapter 7.

4.5.1 Construct Validity

The term *Construct validity* refers to the degree to which the variables, used in the experimental design, actually represent the variables of interest. In this research, it is believed that computational running time correctly represents the efficiency of the method under investigation and that dataset sizes represent the factor that varies the most according to the user's requirements. However, there are other influencers that might have significant effects on the measured outcome, such as system-, hardware- and transport-level variables. Such factors are considered to be mitigated by performing all measurements on the same dedicated machine and using the built in transportation mechanisms of the device.

4.5.2 Internal Validity

Internal validity of an experiment is questioned when the factors involved in the research have the potential of being under the influence of another external factor. Similar aspects arise, as when discussing construct validity: Other influencers such as system-, hardware- and transport-level variables could have an effect on, e.g. the two protocols being compared. However, it is believed that such influencers, in this case, are an inherent part of the nature of the protocols and are, therefore, definitely taken into consideration over the duration of the experiment. The other independent factor used in the experiment, dataset sizes, are set to fixed variables and are considered uninfluenced by internal validity threats.

4.5.3 External Validity

When analyzing the *external validity* of an experiment, the potential of generalizing its findings are explored. In this experiment, it is argued whether an ad-hoc PSI method can outperform a general threshold private set intersection. This thesis has the potential of being used as either qualitative or quantitative measure in future experiments or statistical analyses. Furthermore, this research can be used to motivate whether ad-hoc solutions can outperform general solutions and what restrictions are set on a model when applying those solutions.

4.5.4 Reliability

Reliability is referred to as the ability to replicate the experiment and retrieve the same or similar result. As discussed in Section 4.3, the routing software Routino was utilized. In order to create routes, Routino first retrieves the most recent locational data from OSM. Therefore, if this experiment should be recreated, the routes will be computed using more recent data. This is considered the greatest reliability threat in this experiment. However, by running all test cases the relative efficiency and effectiveness are believed to be correctly represented.

5

Introduction of Time

As of yet, the time of a ride has not been considered as a factor in the O-PrivatePool model. In this chapter, an extended version of O-PrivatePool is presented, where the nodes along a route are given an additional coordinate on a temporal scale. Consequently, a new threshold parameter that represents the degree to which a user is willing to deviate from the time of their ride, is introduced and added to the model. This version of the model is identified as Time-extended PrivatePool (T-PrivatePool). By adding the temporal parameter and threshold to the model, its ability to imitate reality is increased and yet another step is taken towards a production-ready application. This chapter follows the same structure as in Chapter 4 and the same methods, as outlined in Section 3.1, are applied here. In addition, analogous efficiency and effectiveness tests are performed on T-PrivatePool.

5.1 Suggestion

Previous models mentioned in this thesis, i.e. PrivatePool and O-PrivatePool, only consider matching operations in terms of spatial coordinates. However, in a realistic context, in order for a rideshare to be considered feasible, two users are required to be at approximately the same place and time. Therefore, it is required that an additional test is included for determining whether a user can be present at a given spatial coordinate at a similar time as another user, by comparing additional temporal coordinates. To accommodate for the addition of time, Definitions 3 and 4 from Section 2.6.2 were reviewed and updated, which is presented more in depth in Section 5.1.1. The following research question from Section 1.2 was of interest during this iteration:

RQ3: *Is it possible to match entities using an additional threshold, representing deviation in time, without significantly impairing the protocol's efficiency?*

Deduced from *RQ3*, the requirements were twofold; to prove that it is, in fact, possible to match two entities using an additional temporal coordinate and threshold, and to examine whether the addition of the threshold impairs T-PrivatePool's efficiency. Given that a proposed solution was considered sufficient to be able to fulfill the first requirement of *RQ3*, a hypothesis was not formulated for that purpose. However, in addition to the feasibility of being able to match entities using an additional threshold, it was critical to examine whether the task of performing the temporal proximity test has a significant effect on the efficiency, the running time, of the protocol's matching methods. Therefore, the following hypotheses were motivated:

H_{02} : The running times are the same for the endpoint-based matching methods of T-PrivatePool and O-PrivatePool,

$$\mu_{T-PP} = \mu_{O-PP}.$$

H_{A2} : The running time of the endpoint-based matching method of T-PrivatePool is significantly higher than the corresponding method of O-PrivatePool,

$$\mu_{T-PP} > \mu_{O-PP}.$$

H_{03} : The running times are the same for the intersection-based matching methods of T-PrivatePool and O-PrivatePool,

$$\mu_{T-PP} = \mu_{O-PP}.$$

H_{A3} : The running times of the intersection-based matching method of T-PrivatePool is significantly higher than the corresponding method of O-PrivatePool,

$$\mu_{T-PP} > \mu_{O-PP}.$$

These hypotheses were consequently subjected to hypothesis testing. The data generated from the hypothesis tests were used as additional motivations when answering the proposed research question. Due to the fact that the new matching methods of T-PrivatePool consider the fundamental aspects of matching according to the time of the ride, which are not included in O-PrivatePool, they are not considered comparable in terms of effectiveness. Therefore, there was no specific requirement for evaluating the effectiveness of T-PrivatePool. Nevertheless, in order to make general assumptions on how such a model might be affected by adding another matching parameter, an effectiveness test was carried out.

5.1.1 Proposed Solution

In accordance with the requirements in the previous section, the definitions given in Section 2.6.2 were reviewed and updated to accommodate for the addition of time. While Definitions 1 and 2 remain unchanged, Definitions 3 and 4 from Section 2.6.2 have been reconstructed and redefined as Definitions 5 and 6, respectively. Definition 5 now defines the length of a segment as the spatial distance between two points, while Definition 6 sets the condition at which a ridesharing opportunity can be deemed feasible. The two new definitions laid the foundation for the new T-PrivatePool model. T-PrivatePool now has two additional independent variables that need to be taken into consideration during the matching operation, the preferred time of the ride (represented on an axis that indicates the temporal dimension) and a deviation limit that dictates the lower and upper limits of the range, which the user can be present at a specific point in space. Thus, Definition 6 indicates that a rideshare is feasible between two users if there exist two subsequent and distinct points along two users' routes that are within either user's deviation limits and the length of the shareable segment exceeds a given threshold. Note that the deviance functions $\mathit{\Delta}_{xy}$ and $\mathit{\Delta}_{time}$ can be arbitrary functions that represent how much an entity is willing to deviate from their route at a specific point in space and time, respectively.

Definition 5 (Segment length) *Given a segment S for some trip through a graph $G = (V, E)$, let $l(S) = \sum_{i=0}^{|S|-2} d(S[i], S[i+1])$, where $S[j]$ is the jth vertex in S and $d_{xy}(p_1, p_2)$ is the Euclidean distance between the two points p_1 and p_2 in a two-dimensional plane.*

Definition 6 (Ridesharing feasibility) *For any fixed threshold t and deviation function Δ_{xy} (deviance in space) and Δ_{time} (deviance in time), given two trajectories P_A and P_B for users A and B in $G = (V, E)$, ridesharing is feasible for A along a segment $S = \{p_s, ..., p_f\}$ of P_B if and only if*
$l(S) > t$ and there exist two points $P_A[i]$ and $P_A[j]$, with $i < j$, such that:

$$d_{xy}(P_A[i], p_s) < \Delta_{xy}^{P_A}(P_A[i]) \wedge d_{time}(P_A[i], p_s) < \Delta_{time}^{P_A}(P_A[i])$$

$$\wedge d_{xy}(P_A[j], p_f) < \Delta_{xy}^{P_A}(P_A[j]) \wedge d_{time}(P_A[j], p_f) < \Delta_{time}^{P_A}(P_A[j])$$

and $d_{time}(p_1, p_2)$ is the distance between the two points p_1 and p_2 in time.

In the scope of this thesis, users are assumed to have constant speed when traversing their routes. This implies that, if a passenger at some time matches with a driver at the origin of the shareable segment, it is implicitly guaranteed that the passenger will reach the end of the shareable segment at an expected time. Consequently, the deviance in time, denoted by $\mathit{\Delta}_{time}$, is considered to be static for every given point along a route. The construction of a dynamic temporal deviance function that takes into account non-constant speed is left for future work. Additionally, the construction of the traversable graph and individual user's trajectories are considered out of scope. The spatial deviance function $\mathit{\Delta}_{xy}$ remains the same, as given by Equation 4.1.

Based on the new definitions, the time-sensitive endpoint-based and intersection-based matching patterns, are presented in the subsequent sections.

5.1.1.1 Endpoint-based Matching

User A, who holds a pair of asymmetric cryptographic keys, makes one key publicly available. Given a threshold of maximum time deviance as τ_{max}, A initiates the proximity test by querying B with the variables $((x_A, y_A, x_A^2, y_A^2), z_A)$. B uses these variables, along with its own values, to compute the euclidean distance between them as:

$$D_{xy} = x_A^2 + y_A^2 + x_B^2 + y_B^2 - 2(x_A x_B + y_A y_B).$$

B then computes the temporal distance as:

$$D_\tau = z_B - z_A$$

Given that these computations performed by B are all linear operations, they can be performed in an encrypted domain, using homomorphic encryption. B can then compare the value D_{xy} with the spatial radius value r, by constructing a set in the following manner:

$$\{(D_{xy} - i)\varrho_i | i = x \times y | x, y \in \{0..r\}\}.$$

Similarly, B constructs the set of temporal comparisons:

$$\{(D_\tau - j)\varrho_j | j \in \{-\tau_{max}...\tau_{max}\}\}.$$

Where r and τ_{max} are generated from their respective deviance functions and each ϱ is an independent random number in the plaintext space. The contents of the sets are then sent to A in a random order. Hence, A can only conclude whether $\exists i < r^2, -\tau_{max} < j < \tau_{max} : i = D_{xy} \wedge j = D_\tau$, which is the equivalent of $D < r^2 \wedge |j| < \tau_{max}$, by multiplying together the values within each set and then adding those values together to acquire the final result. If the result is a zero value, it has been deduced that the two parties are closer together than the value r. If the distance between the two users is not within the limits, the resulting value will be a random non-zero value.

5.1.1.2 Intersection-based Matching

As with the O-PrivatePool intersection-based matching method, it is required that the two compared datasets are sorted. Furthermore, it is given that Alice has a trajectory T^A of size m and Bob a trajectory T^B of size n, for any given threshold t. Additionally, T_{xy_i} is defined as the spatial coordinate at the ith point along the trip T. Similarly, $T_{\tau_{i,p}}$ is defined as the pth possible temporal coordinate at the ith point along the trip T. Moreover, θ is defined as the fixed interval size between two given temporal coordinates at an arbitrary spatial coordinate. The parameter k is then defined as the number of time slots available for that point. To determine the value of k, Equation 5.1 is applied:

$$k = 1 + \frac{\tau}{\theta} \tag{5.1}$$

Logically, if the driver's velocity is assumed constant and there exist a point on both routes, whose temporal and spatial coordinates fall within the deviation limits of the driver, then it is guaranteed that the passenger will reach all other consequent points along the shared segment in due time. Therefore, we need that if

$$\exists i, j, p, q : T^A_{xy_i} = T^B_{xy_j} \land T^A_{xy_{i+t}} = T^B_{xy_{j+t}} \land T^A_{\tau_{i,p}} = T^B_{\tau_{j,q}},$$

then it also holds that

$$\forall u \in \{i, ..., i+1\}, v \in \{j, ..., j+t\}, \exists p \in \{1, ..., k^A\}, \exists q \in \{1, ..., k^B\} : T^A_{xy_u} = T^B_{xy_v} \land T^B_{\tau_{i,p}} = T^B_{\tau_{j,q}},$$

$$\forall u \in \{0, ..., t\}, \exists p \in \{1, ..., k^A\}, q \in \{1, ..., \theta^B\} : T^A_{xy_{i+u}} = T^B_{xy_{j+u}} \land T^A_{\tau_{i+u,p}} = T^B_{\tau_{j+u,q}}.$$

The time-sensitive intersection-based method is outlined as follows:

1. Alice prepares a set on the following form,

$$(T^A_{xy_1}, T^A_{xy_t}, T^A_{\tau_{1,1}}), (T^A_{xy_1}, T^A_{xy_t}, T^A_{\tau_{1,\theta}}), ..., (T^A_{xy_1}, T^A_{xy_t}, T^A_{\tau_{1,k}}),$$
$$(T^A_{xy_2}, T^A_{xy_{t+1}}, T^A_{\tau_{2,1}}), (T^A_{xy_2}, T^A_{xy_{t+1}}, T^A_{\tau_{2,\theta}}), ..., (T^A_{xy_2}, T^A_{xy_{t+1}}, T^A_{\tau_{2,k}}),$$
$$...,$$
$$(T^A_{xy_{m-t+1}}, T^A_{xy_m}, T^A_{\tau_{m-t+1,1}}), (T^A_{xy_{m-t+1}}, T^A_{xy_m}, T^A_{\tau_{m-t+1,\theta}}), ..., (T^A_{xy_{m-t+1}}, T^A_{xy_m}, T^A_{\tau_{m-t+1,k}})$$

2. Bob similarly prepares a set on the form,

$$(T^B_{xy_1}, T^B_{xy_t}, T^B_{\tau_{1,1}}), (T^B_{xy_1}, T^B_{xy_t}, T^B_{\tau_{1,\theta}}), ..., (T^B_{xy_1}, T^B_{xy_t}, T^B_{\tau_{1,k}}),$$
$$(T^B_{xy_2}, T^B_{xy_{t+1}}, T^B_{\tau_{2,1}}), (T^B_{xy_2}, T^B_{xy_{t+1}}, T^B_{\tau_{2,\theta}}), ..., (T^B_{xy_2}, T^B_{xy_{t+1}}, T^B_{\tau_{2,k}}),$$
$$...,$$
$$(T^B_{xy_{m-t+1}}, T^B_{xy_m}, T^B_{\tau_{m-t+1,1}}), (T^B_{xy_{m-t+1}}, T^B_{xy_m}, T^B_{\tau_{m-t+1,\theta}}), ..., (T^B_{xy_{m-t+1}}, T^B_{xy_m}, T^B_{\tau_{m-t+1,k}})$$

3. The parties run a standard PSI protocol on these sets.

5.1.2 Experimental Design

In this chapter, two pairs of hypotheses are presented. Therefore, two independent hypothesis tests were designed. The experimental design for each respective hypothesis test is presented in Sections 5.1.2.1 and 5.1.2.1, followed by an experimental design for the supplementary effectiveness test.

5.1.2.1 Endpoint-based Matching

The experimental design presented here was constructed for the purpose of testing hypotheses H_{02} and H_{A2}, where H_{A2} indicates an underlying suspicion that the running times of the endpoint-based method of T-PrivatePool will be significantly higher than that of O-PrivatePool. Therefore, it was determined to apply a one-tailed hypothesis test, giving a corresponding α level as 0.025.

Due to resource constraints, it was determined that it would suffice to test the effects of adding an additional threshold parameter on the method by measuring the running times while either applying the threshold parameter or not. Therefore, the sole factor involved in this experiment was the usage of a time parameter when matching and its levels set to either *true* or *false*. Given this information, a simple one-factor experimental design, with paired comparisons was deemed appropriate. It was determined that 25 measurements per factor level would suffice in reducing any noise in the experiment.

5.1.2.2 Intersection-based Matching

For the final two hypotheses, H_{03} and H_{A3}, it was determined to reuse pre-existing test suites, from Chapter 4, with minor modifications. That decision lead to the same experimental design being applied, with three total factors. The factors involved in this experiment are the following: *dataset size*, *maximum time deviation* and *time deviation precision*. The *dataset size* indicates the number of nodes within each route being compared. *Maximum time deviation* determines how much the user is willing to deviate in time, from any given point along their route, in terms of minutes. The *time deviation precision* factor determines the intervals, which the maximum deviation should be divided into, given in minutes.

Due to the biased nature of the alternative hypothesis, H_{A3}, a one-tailed test was applied. Additionally, due to the infinite number of possible levels for each factor in the experiment, a *Fractional Factorial* design was applied, with the α level as 0.025. The experimental unit, in this case, was the intersection-based method of T-PrivatePool, while the response variable was the running time of the method. The *dataset size*, as in Chapter 4, had eight levels, i.e. the integer values 5 through 12. The levels of the *dataset size* factor indicate the exponent value n, which was used to compute the actual size 2^n. *Maximum time deviation* was restricted to six levels ranging from 0 to 12 hours, with 2 hour intervals. The motivation behind the determined range, is that users generally search for available rides the same day as they intend to make use of it. The levels of time deviation precision were set to 30, 45 and 60 minutes. Since the number of possible treatments were still considered large, the number of replications were limited to 10 per treatment.

5.1.2.3 Effectiveness

The effectiveness of T-PrivatePool was measured by independently applying the endpoint- and intersection-based matching patterns, with an additional temporal proximity test. In the temporal proximity test, the two points that are investigated may not exceed the temporal distance value d. The possible values of d are set as 30, 45 and 60 minutes. The maximum spatial distance, determined by r, and the minimum shareable threshold value, determined by t, are defined in the same manner as for Chapter 4. The results from the two effectiveness tests are then combined and compared to the total number of ridesharing opportunities, found by implementing and applying the brute force algorithm as outlined in Definition 6.

5.1.3 Development

The development phase was divided into three parts; the first two parts involved extending O-PrivatePool to include the algorithms given in Section 5.1, while the third part involved adding additional proximity tests to the effectiveness test suite. The extended O-PrivatePool model with additional variables for temporal matching is referred to as T-PrivatePool.

Initially, a point along a user's route was redefined. A point was given an additional dimension, so that it does not only refer to a point in space but also in time. T-PrivatePool was then made to export the route according to the first two steps of the new intersection-based matching pattern, in Section 5.1.2.2. Each triplet represents each independent time at which the user is willing to be present at a given spatial coordinate:

$$(T^A_{xy_1}, T^A_{xy_t}, T^A_{\tau_{1,1}}), (T^A_{xy_1}, T^A_{xy_t}, T^A_{\tau_{1,\theta}}), ..., (T^A_{xy_1}, T^A_{xy_t}, T^A_{\tau_{1,k}}),$$

$$(T^A_{xy_2}, T^A_{xy_{t+1}}, T^A_{\tau_{2,1}}), (T^A_{xy_2}, T^A_{xy_{t+1}}, T^A_{\tau_{2,\theta}}), ..., (T^A_{xy_2}, T^A_{xy_{t+1}}, T^A_{\tau_{2,k}}),$$

$$...,$$

$$(T^A_{xy_{m-t+1}}, T^A_{xy_m}, T^A_{\tau_{m-t+1,1}}), (T^A_{xy_{m-t+1}}, T^A_{xy_m}, T^A_{\tau_{m-t+1,\theta}}), ..., (T^A_{xy_{m-t+1}}, T^A_{xy_m}, T^A_{\tau_{m-t+1,k}})$$

Given that O-PrivatePool was written in Python, the same programming language was used during the development of T-PrivatePool. The temporal proximity test was implemented according to the proposed solution in Section 5.1.1.1. To measure the running times of the model, an automatic timer was made to record the elapsed time from when the endpoint-based matching method was called, until the method finished its computation.

To make T-PrivatePool compatible with BaRK-OPRF, the PSI protocol was changed so that the exported data from T-PrivatePool is parsed correctly. This change was made in the programming language C++, using Microsoft Visual Studio. The efficiency test suite for the intersection-based matching method, comprises of the same tasks as were performed in the test suite for Chapter 4. However, the datasets were randomly generated according to the new method outlined in Section 5.1.1.2.

The effectiveness test, that was used in Chapter 4, was altered according to the prescribed changes given in Section 5.1.2.3.

5.1.4 Evaluation

Initially, a worst-case time complexity analysis was carried out. The results of this analysis are displayed in Section 5.2. Both efficiency tests were moved to the dedicated machine and executed uninterrupted. The total number of replications were 25 and 10 for endpoint-based matching and intersection-based matching, respectively, with factors set to the levels specified in their corresponding experimental designs.

The routes used to evaluate the efficiency of the endpoint-based method were generated from two starting points on different ends of Chalmers University of Technology's Johanneberg campus, Doktor Forselius Backe 17 and Maskingränd 2, while the endpoints were set to Chalmers' Lindholmen campus. For this experiment, τ was fixed to 15 minutes, with interval sizes $\theta = 1$ minute. Using Equation 5.1, the total number of time slots for each point along the path, k, was determined to be 15. Measurements from the endpoint-based method's efficiency test can be found in Appendices B and C.

As mentioned in Section 5.1.3, the datasets for evaluating the intersection-based method's efficiency were randomly generated, while the data given by TLC was used to evaluate the effectiveness of model. More specifically, 1000 rides from every even month of 2015 were used, in order to be analogous to the effectiveness test used in Chapter 4.

The full list of measurements from the intersection-based method's efficiency tests can be found in Appendix D. The regression analyses performed on the efficiency data, using the regression tool SGS, are presented in Section 5.2.1. Results of the effectiveness tests and their analysis is presented in Section 5.2.2.

5.2 Results

Before presenting the statistical analyses, the worst-case algorithmic complexity of the individual solutions are evaluated.

The solution given in Section 5.1.1.1, indicates that all operations of the algorithm, until the construction of the masking sets, can be performed in constant time. The aforementioned is true, given that the complexity of the homomorphic encryption method is excluded from the analysis. Furthermore, the construction of the masking set of the spatial proximity test is performed in $\mathcal{O}(n^2)$ time, where n is the number of possible distance values from the user. However, the masking set of the temporal proximity test can be constructed in $\mathcal{O}(m)$ time, where m is the number of possible temporal deviation values from the user. Furthermore, the individual proximity tests are performed in sequence and mark the procedures with the highest degree of complexity compared to all consequent operations. Therefore, the algorithm is determined to be bounded by the highest exponent:

$$\mathcal{O}(n^2 + m) = \mathcal{O}(n^2).$$

Note that, when analyzing the complexity of the intersection-based method of T-PrivatePool, the complexity of the PSI method of choice is abstracted away, as previously elaborated in Chapter 4. According to the solution given in Section 5.1.1.2, the construction of the two datasets in the first two steps can be carried out in two intermediate tasks. The first task consists of looping linearly through the points at the first $n - t + 1$ indices and pairing them with the corresponding point at the indices t through $n+1$. The pairing activity consists of simple constant operations. The second task consists of running through all available times m for each pair and adding the given temporal coordinate to the pair, constructing an m number of triplets. Both intermediate tasks are performed in nested loops and, therefore, the worst-case complexity of the first two steps are given as

$$\mathcal{O}(n \times m).$$

The results of the regression analyses of the measurements in efficiency are presented in Section 5.2.1. Effectiveness results are presented in the succeeding section, Section 5.2.2.

5.2.1 Statistical Analysis of Efficiency

Given that there were two separate experiments that were run in parallel, two sets of statistical results were generated. Statistical analyses of the endpoint-based method's measurements presented in Appendices B and C are displayed in Section 5.2.1.1, while statistical analyses of the intersection-based method's measurements in Appendix D are presented in Section 5.2.1.2. Each table presented in the subsequent sections will not be explained in great detail, since they resemble the tables presented in Section 4.4.1.

5.2.1.1 Endpoint-based Matching

By applying SGS's *Regression* feature on the data given in Appendices B and C, possible coefficients for the prediction model, given by Equation 4.2, were generated. The coefficient table that represents the change in efficiency when utilizing a time and maximum time deviance parameters are displayed in Table 5.1. Note that according to Table 5.1, both the *Intercept* coefficient and the coefficient *Time*, indicating the application of a time deviation parameter, both have *p values* under the given α level 0.025. In Table 5.2, the overall goodness of the fit of the model, suggested by SGS is presented. The parameter *R-square*, which describes how much variation in the measured value can be explained by the model, is given as 98.7%. Additionally, the *Adj R-square* is given as 98.6%.

Name of coefficient	**Estimate**	**Std. Error**	**t value**	**p value**
Intercept	0.382	0.013	29.908	<0.0001
Time = true	1.076	0.018	59.513	<0.0001

Table 5.1: Coefficients and their corresponding standard error, t and p values generated by running SGS on the running times of an endpoint-based matching with time included

Variable	**Value**
R-square	0.987
Adj R-square	0.986
Residual SD	0.064
Sample SD	0.547
N observed	50
N missing	0

Table 5.2: Parameters indicating the overall fit of the prediction model for endpoint-based matching with time

The *ANOVA* table for the prediction model is presented in Table 5.3. It can be observed that the *F value* of the model is considerably high, giving the impression that the model explains a large amount of the variation in the observed measurement. Additionally, the *p value* is shown to be lower than the set α level of 0.025.

ANOVA Table					
Factors	**Df**	**Sum Sq**	**Mean Sq**	**F value**	**p value**
Model	1	14.471	14.471	3 541.750	<0.0001
Residual	48	0.196	0.004		
Total	49	14.667	0.299		

Table 5.3: ANOVA table derived from the measured data in Appendices B and C

In Table 5.4, the factor of applying a comparison of time coordinates is subjected to a *partial F test*. A *partial F test* can help to determine how much of the variance in the response variable can be explained by the individual factors. In this case, the F and *p values* are the same as in Table 5.3, given that it is the only factor considered in the experiment. Special attention is brought to the *p value*, which is lower than the α level 0.025.

Partial F Tests					
Factors	**Df**	**Sum Sq**	**Mean Sq**	**F stat**	**p value**
Time	1	14.471	14.471	3 541.750	<0.0001

Table 5.4: Partial F tests for individual factors included in the prediction model

5.2.1.2 Intersection-based Matching

By applying SGS's *Regression* feature on the efficiency measurements of the intersection-based method, given in Appendix D, possible coefficients for the prediction model are generated. The coefficient table is given in Table 5.5. In can be observed that the *Intercept* coefficient, along with all the coefficients representing the independent factors have *p values* lower than the α level 0.025.

Name of coefficient	**Estimate**	**Std. Error**	**t value**	**p value**
Intercept	0.089	0.005	19.308	<0.0001
Trajectory size	0.000	0.000	35.394	<0.0001
Max time deviation	0.004	0.000	14.868	<0.0001
Time deviation precision	-0.001	0.000	-6.091	<0.0001

Table 5.5: Coefficients and their corresponding standard error, t and p values generated by running SGS on the running times of an intersection-based matching with time included

Data representing the overall fitness of the prediction model is given in Table 5.6. *R-square* indicates that the model explains 47.7% of the variance in running times of the intersection-based matching method. However, *Adj R-square* indicates that the model can account for 47.3% of the variance in the response variable.

Variable	**Value**
R-square	0.474
Adj R-square	0.473
Residual SD	0.046
Sample SD	0.063
N observed	1 680
N missing	0

Table 5.6: Parameters indicating the overall fit of the prediction model for intersection-based matching with time

The *ANOVA* table for the prediction model is presented in Table 5.7. As in the previous section, the *F value* of the model is considerably high. It should also be noted that the *p value* is lower than the α level 0.025.

ANOVA Table					
Factors	**Df**	**Sum Sq**	**Mean Sq**	**F value**	**p value**
Model	3	3.142	1.047	503.625	<0.0001
Residual	1676	3.486	0.002		
Total	1679	6.628	0.004		

Table 5.7: ANOVA table derived from the measured data in Appendix D

5.2.2 Analysis of Effectiveness

The results of testing the effectiveness of the T-PrivatePool protocol are presented in Table 5.8. The values in the column *IS* represent the average percentage of matching opportunities detected by the updated intersection pattern compared to a brute force approach. The values in the column *EP* represent the percentage of opportunities detected by the endpoint-based matching pattern. The brute force approach compares each individual point along both respective routes, while applying varying restriction parameters. The column *T-PP* combines the efforts of the individual patterns. The restriction parameters include the threshold t and r_0 from the effectiveness tests in the previous chapter, with an additional d parameter that represents the maximum temporal distance between two points, in minutes. The spatial radius value of intermediate points are, as described in Equation 4.1, dynamically computed using the value r_0 and the index of the point.

		$r_0 = 500$			$r_0 = 1000$			$r_0 = 2000$		
t	d	T-PP	IS	EP	T-PP	IS	EP	T-PP	IS	EP
20%	30	88.84	0.00	88.84	60.51	0.76	59.75	35.23	3.63	31.60
	45	94.69	1.42	93.26	62.34	1.39	60.95	35.38	3.47	31.91
	60	96.92	1.03	95.90	64.97	1.36	63.61	37.31	3.97	33.34
50%	30	76.58	0.00	76.58	42.97	1.52	41.46	22.03	6.31	15.72
	45	79.96	3.81	76.15	44.48	2.89	41.59	21.35	5.84	15.51
	60	79.33	3.12	76.22	45.93	3.03	42.90	23.68	6.95	16.72
80%	30	34.03	0.00	34.03	11.79	2.38	9.40	9.58	7.28	2.30
	45	42.87	9.72	33.14	16.23	5.03	11.21	9.80	7.00	2.81
	60	39.55	7.04	32.51	16.07	4.97	11.10	11.23	8.24	2.99

Table 5.8: Average percentages of detected ridesharing opportunities in T-PrivatePool

5.3 Threats to Validity

This chapter addresses potential threats to validity concerning the experiments carried out when measuring the efficiency and effectiveness of T-PrivatePool. By addressing the experiment's validity threats and discussing their mitigations, the trustworthiness of the results presented in Section 5.2 is believed to be reinforced. Furthermore, the contents of this section is taken into consideration when discussing the results in Section 7.2.

5.3.1 Construct Validity

Whenever measuring the change in efficiency in the endpoint-based matching method, it is a possibility that not all relevant factors were taken into consideration. Due to time constraints, the test suites were limited to a one-factor design. Other possible factors, such as varying levels in temporal deviance can affect the model. By keeping the range of possible temporal coordinates relatively low it is believed that, if the factor has a significant effect on the response variable, its influence should still appear systematically in the measured value.

The same can be stated for the intersection-based matching, that additional factors should certainly be taken into account. This suspicion is founded on the basis that, according to the *Adj R-square* in Table 5.6, the model can not account for more than half of the variation in the response variable. Nevertheless, the *p value* in Table 5.7 indicates that the model can, indeed, explain a significant amount of the variation. This will be explored further in Section 7.2.2.

5.3.2 Internal Validity

Similar to the internal validity threats as in Chapter 4, additional sources of variation such as system-, hardware- and transport-level variables could have an effect on the measurements. However, this is believed to be effectively mitigated by the usage of almost identical test cases and the same simulation setup.

5.3.3 External Validity

It is believed that the findings can be generalized, such that the effects of incorporating additional matching factors on the running times of current privacy-preserving models can be observed. More precisely, given that it is a fundamental challenge of software development to make tradeoffs between functionality and efficiency, the results of this thesis can help give the reader an intuition of how efficiency is affected when applying supplementary functionality to a privacy-preserving applications.

5.3.4 Reliability

The test suites used for measuring the effectiveness of the model suffer from the same drawback, in terms of replication, as in Chapter 4. I.e. the results are depended on the current version of the OSM data. Therefore, if this experiment should be recreated, the routes will be computed using more recent data. However, this threat can be mitigated by running all test cases locally, since the results are believed to give the same relative changes in efficiency and effectiveness, independent of the time of the experiment's execution.

6
Authentication of End-users

In previous chapters, it is generally assumed that the users of the model are both aware of each other and that the other party is an authenticated user. However, neither has a secure authentication method been suggested nor implemented. In this chapter, the option of introducing an authentication method for end-users is investigated, that could additionally be used to identify the users. As mentioned in Section 1.3, all research regarding the authentication of end-users was open ended. Given that, there was neither a research question to answer nor any commitment to implementing the authentication methods, there was no need for a statistical hypothesis test. Therefore, the structure of this chapter differs from the previous ones.

An investigation was carried out on two common authentication systems built on cryptography that are used in distributed and decentralized systems. Based on their pros and cons, an evaluation was made to ascertain what type of authentication system is preferred to utilize in an application, such as PrivatePool. An application with many distributed users face challenges such as choice of cryptographic algorithms, synchronization and the amount of trust that has to be placed on a third party. An authentication protocol builds on cryptographic paradigms and the two most common ones are symmetric key and asymmetric algorithms.

6.1 Exploring Authentication Methods

Traditional authentication systems tend to store users' knowledge factors, such as username and password, in a centralized manner. Such centralized authentication systems are generally assumed to be trusted. However, it cannot be expected that its security is unbreachable. In 2017, a password manager and a single sign-on provider OneLogin suffered a breach of security [48], where the attackers had the ability to decrypt their encrypted data. PrivatePool, its successors and other privacy-preserving applications are no exceptions, should they be implemented with authentication systems. Therefore, recommendations are provided for two different authentication technologies for distributed and decentralized systems that, either to some extent or completely, remove the the need for a centralized entity.

6.2 Kerberos

Kerberos is a well known authentication protocol, developed by Massachusetts Institute of Technology, that is used in *distributed systems* and depends on the existence of a dedicated and trusted authentication server [34]. Kerberos builds on symmetric key cryptography, where a dedicated server handles the security aspects and is in charge of user and session authentication, name and password administration. The objective is to make it possible for nodes that communicate over a non-secure network to prove their identity to others in a secure manner.

6.3 Blockchain

The software platform for digital assets, or blockchain, is a *decentralized system* that allows digital information to be distributed and tracked without the involvement of third parties [21]. The blockchain is a digital distributed ledger which can be seen as a database that holds a list of records which are linked together and secured with cryptography. A blockchain is considered immutable and tamper-proof when it is shared among a large number of hosts. It is updated independently by each participant (node) in a large network. Opposite to a traditional client-server architecture, updates are not communicated to various nodes by a central entity. Rather, all updates within the network are broadcasted to and processed by every single node, which adds the update to their blockchain and submits its chain as the most recent one. All other nodes perform the same task and vote for their version of the chain. After all votes have been received, a consensus is reached among the nodes through majority voting.

The decentralized nature of the blockchain network and its transparency characteristic provides the opportunity to develop a secure identity management system. However, instead of leveraging the blockchain as a database, it can be used as a independent source of truth for identity certification. Since the blockchain holds the property of being immutable, it is resistant to attacks such as information leakage, modification or deletion. By either utilizing an identity management application or having it integrated in the ridesharing application, users can store their personally identifiable information (PII) on their personal mobile device, which they manage and share on their own terms. PII can be represented as biometrics like fingerprints and iris patterns. For the sake of clarification, the PII is independently verified with one way digital signatures of hashes and the blockchain holds no PII. Identity certifications are instead stored on the blockchain to use them to independently verify a user and with the help of leveraging hashes, salts, and digital signatures, it is not possible to reverse engineer them to their original form.

7

Discussion

Discussions for the result of each individual iteration are carried out in their corresponding sections of this chapter. For every section, the findings presented in the *Results* section of Chapters 4 and 5 are reviewed and evaluated. The findings are put into context and consequently used to evaluate relevant hypotheses and answer the research questions presented in Section 1.2. Additionally, an evaluation of the authentication methods presented in Chapter 6 is carried out.

7.1 Matching Operation

According to the time complexity analysis carried out in Section 4.4, the worst-case time complexity of O-PrivatePool's intersection-based matching method is determined as $O(n)$. With T-KEM's complexity given as $O(n^3)$, the first hypothesis, H_{01}, is rejected in favor of the alternative hypothesis, H_{A1}. The running time of O-PrivatePool's intersection-based matching method is significantly lower than PrivatePool's method,

$$\mu_{O-PP} < \mu_{PP}.$$

Furthermore, the results given in Section 4.4 show that T-KEM performs better for dataset sizes 32 an under, then the method's running times continue to grow according to a cubed function. The altered version of BaRK-OPRF, however, seems to produce running times that follow an almost constant trend, maintaining a running time of approximately 0.1 second independent of the input size for section numbers 4096 and under. This was, therefore, interpreted as a preliminary indication that the intersection-based matching method of O-PrivatePool performs generally better than the intersection-based matching method of PrivatePool.

To confirm these results, we look to the statistical analyses. Results of the *ANOVA* test in Table 4.4 show that the prediction model, which applies the coefficients given in Table 4.2, has a *p value* lower than 0.0001. Considering that the *p* value is lower than the set α value (0.025), it was determined that the model can account for a statistically significant amount of the variation in the running times. Hypothesis H_{02} was, therefore, rejected in favor of the alternative hypothesis, H_{A2}, with 97.5% confidence. Furthermore, it can be determined from the partial *F tests* in Table 4.5 that, since the *p values* of both factors (the size of the dataset and the protocol used) are below α, both factors have a statistically significant effect on the running times. Additionally, the *Adj R-squared* value in Table 4.3 shows that 47.2% of the variance in the running times can be described by the varying levels of both factors.

Given that the results in Tables 4.6 and 4.7 are identical, it can be determined that there is exactly no loss in effectiveness when applying the new intersection-based method. Additionally, the threats discussed in Section 4.5 are assessed to be sufficiently mitigated and, therefore, are not assumed to have any particular effect on the results of this experiment.

7.2 Introduction of Time

The discussion of the effects of the additional matching parameter, the time of the ride, on T-PrivatePool's efficiency is carried out in two separate sections. The former takes the resulting effects of adding time and time deviance to the endpoint-based method of O-PrivatePool into consideration, while the latter discusses the effects of adding the same parameters to the intersection-based method. Following the discussion of change in efficiency, the effectiveness of the T-PrivatePool model is examined.

7.2.1 Endpoint-based Matching

By reading Table 5.1, it can be seen that the *p value* of both coefficients, the *Intercept* and *Time*, are below the α level of 0.025. Therefore, they have a significant effect on the model. Furthermore, according to the *ANOVA* table in Table 5.3, the *p value* of the model is also below the α level. This means that the model can account for a significant amount of the variance in the response variable. Taking that into consideration, hypothesis H_{03} is rejected in favor of the alternative hypothesis, H_{A3}, with 97.5% confidence.

Table 5.2 shows that the model can account for 98.6% of the variance, given by the *Adj R-square* value. Furthermore, the *partial F test* given in Table 5.4 provides additional evidence that the *Time* factor does have a significant effect on the response variable. This is determined by the fact that the *p value* is lower than the given α level.

The greatest threat to the validity of this experiment is that the only factor that was involved was the binary variable of whether a temporal deviation threshold was used in the computation of the endpoint-based matching or not. The problem is that the computation itself is reliant on other factors, such as the varying levels of possible temporal deviance and the precision to which the deviance is divided into. However, it is believed that all of these factors have been correctly combined and isolated by restricting the deviance and deviance precision to fixed and relatively small values.

7.2.2 Intersection-based Matching

By reading the *p values* of the coefficients in Table 5.5, it is evident that all individual coefficients can do a useful amount of explaining within the model, given that they are all below the α level 0.025. Additionally, according to the *ANOVA* table in Table 5.7, the *p value* is below the α, which indicates that the model can account for a statistically significant amount of the variance in the running times of the intersection-based matching method. More precisely, according to the *Adj R-square* value in Table 5.6, the model can account 47.3% of the variance. Hypothesis H_{03} is, therefore, rejected in favor of the alternative hypothesis, H_{A3}, with 97.5% confidence.

The most notable drawback of this experiment, as mentioned before in Section 5.3.1, is that *Adj R-square* in Table 5.6 indicates that the prediction model can only account for 47.3% of the variance in the running times. However, the fact that the *p value* of the model, seen in Table 5.7, is lower than α suffices to relieve that suspicion. Despite that fact, it would be interesting for future work to investigate and include other potential factors that could be accountable for the rest of the variance.

7.2.3 Effectiveness

From the column *T-PP* in Table 5.8, it can be observed that the matching methods of T-PrivatePool are able to detect a relatively high percentage of the total matching opportunities, when r_0 is restricted to 500 and t is restricted to 20%. However, the proportion of detected opportunities decrease rather dramatically when the spatial deviance threshold is extended and the minimum intersection threshold is extended. This total percentage holds hand in hand with the opportunities detected by the endpoint-based matching method. However, the opposite trend seems to be for the intersection-based method. Furthermore, for both matching methods, the percentage of detected opportunities generally increase as the temporal deviance threshold is increased. The usefulness of the intersection-based method can be argued at this point, given that it the relative number of detected ridesharing opportunities only increase substantially when the conditions of the opportunity are severely restricted.

7.3 Authentication of End-users

By comparing the two authentication systems, it can be seen that a Kerberos system is still dependent of a central server. This type of architecture is still prone to single points of failure and central repositories that need to maintain credentials such as usernames and passwords, which makes it vulnerable to breaches by attackers. On the other hand, utilizing a blockchain will eliminate a centralized server and its downsides. However, as mentioned in Section 6.3, every single node runs the blockchain in order to reach consensus. This could be considered wasteful, since it is slower and more expensive, compared to a traditional single computer.

8

Conclusion

This chapter presents conclusions based on the results in the preceding chapters. Subsequently, the perspective is moved away from the ridesharing context to a broader view of the findings.

Following the discussion in Section 7.1, it is clear from the worst-case complexity analysis and the rejection of H_{01} and H_{02}, that the research questions *RQ1* and *RQ2* are both answered positively. This means that the new proposed O-PrivatePool's intersection based matching method has both less time complexity than PrivatePool's corresponding method based on T-KEM and lower running time when running benchmarking tests. If we were to generalize these findings it could be said that, by applying ad-hoc solutions, it is definitely possible to decrease the overhead of privacy-preserving models. However, the hope for the future is that SMC protocols can perform well enough to be considered at production-level quality, while retaining a great deal of generality such that they can be applied to a wide range of contexts.

The results and discussions of Chapter 5 show that both null hypotheses in this iteration were rejected. Therefore, we conclude that by applying additional parameters, the efficiency of the method is impaired. The research question *RQ3* is, thus, answered negatively: It is not possible to match entities using an additional threshold, representing deviation in time, without impairing the protocol's efficiency. However, since the introduction of additional matching parameters can be seen as additional work for the method to carry out, this behaviour was expected. Generally, the effects of adding more work for the privacy-preserving methods to carry out seem to increase their running times to a certain degree. Although, the running times are still relatively low and further research could reveal whether they are within the actual acceptance limits of users.

The research during the authentication phase revealed that, since the desire is to eliminate the third party centralized system, which handles their customer's data, we estimate that a Kerberos system would not be optimal and go against the purpose of a decentralized system. It should, however, be noted that blockchain is still a very new technology and it does present quite a few challenges that need to be overcome before it can be used in practical applications.

In today's increasingly digital world, awareness and concerns related to data privacy and security is growing and becoming more critical. General Data Protection Regulation (GDPR) is being implemented to protect the data of EU citizens from data breaches and other form of misuse [27]. Thus, PSI could perhaps be a possible solution and make it possible to still find intersections of sets that now consists of private data.

Even though SMC protocols (such as PSI) suffer from additional computational overhead compared to their naïve counterparts, the techniques used in the field are constantly decreasing the divide and moving their capabilities closer to something that could be considered production level. Despite the fact that language-based security is gaining more traction, more attention is needed on other levels of security. Information such as the number of requests to the open source mapping software, their frequencies and their response sizes (to take OSM as an example) can still reveal a significant amount of information.

8.1 Future Work

The models presented in this thesis do not take factors, such as the varying lengths of the segments in a route or that entities travel their routes with a non-constant velocity, into consideration. By taking these factors into consideration, users would be able to match with each other according to the respective route's length and a more precise match could be made according to a non-constant temporal deviation function. Furthermore, in reality, there is a fundamental relationship between deviation in time and deviation in space. If a user were to deviate in time, the tolerance for deviating any further in space should change accordingly. This relationship can potentially be expressed as a combined function of the two devation functions mentioned in this thesis, by e.g. represent it as an ellipsoid. When computing the radius value along the temporal axis, we estimate that a left-modal curve would best suit a driver who is restricted to be present at a specific time at the end of their ride. Otherwise, if the driver is restricted to depart from their origin at a specific time, we estimate that a right-modal curve would be best suited. Should the passenger be restricted in a similar manner, an additional check could be made to ensure that both parties depart at times which fall within their limitations. Time deviance functions could also be used to reflect an estimated encounter with congestion or other changes in traffic.

Further investigative research needs to be carried out into what possible authentication methods are available for distributed and decentralized systems, that take privacy preservation into consideration. Currently, there is no implementation of an authentication method in the models that we have presented. However, that would require the previously mentioned investigative research to be carried out.

Bibliography

[1] A. M. Bishop. Routino : Router for OpenStreetMap Data. `https://www.routino.org/`.

[2] BlaBlaCar. Share your journey with BlaBlaCar - Trusted carpooling. `https://www.blablacar.com/`.

[3] R. Carlton, A. Essex, and K. Kapulkin. Threshold Properties of Prime Power Subgroups with Application to Secure Integer Comparisons. Cryptology ePrint Archive, Report 2018/224, 2018. `https://eprint.iacr.org/2018/224`.

[4] Microsoft Corporation. Visual Studio IDE, Code Editor, VSTS, & App Center. `https://www.visualstudio.com/`.

[5] I. Damgørd, M. Geisler, and M. Krøigard. Homomorphic Encryption and Secure Comparison. *Int. J. Appl. Cryptol.*, 1(1):22–31, February 2008.

[6] T. Elgamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, 31(4):469–472, Jul 1985.

[7] C. Fontaine and F. Galand. A Survey of Homomorphic Encryption for Non-specialists. *EURASIP Journal on Information Security*, 2007(1):013801, Dec 2007.

[8] OpenStreetMap Foundation. OpenStreetMap. `https://www.openstreetmap.org/`.

[9] M. J. Freedman, K. Nissim, and B. Pinkas. Efficient Private Matching and Set Intersection. In C. Cachin and J. L. Camenisch, editors, *Advances in Cryptology - EUROCRYPT 2004*, pages 1–19, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

[10] D. Freni, C. R. Vicente, S. Mascetti, C. Bettini, and C. S. Jensen. Preserving Location and Absence Privacy in Geo-social Networks. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, CIKM '10, pages 309–318, New York, NY, USA, 2010. ACM.

[11] Google Inc. Google Scholar. `https://scholar.google.se/`.

[12] V. Grover. A Tutorial on Survey Research: From Constructs to Theory. *Journal of Operations Management*, 16(4), 2001.

[13] P. Hallgren. *Robust location privacy.* Doctoral dissertations at Chalmers University of Technology. Series, no: 4286. Department of Computer Science and Engineering, Chalmers University of Technology, 2017.

[14] P. Hallgren, M. Ochoa, and A. Sabelfeld. InnerCircle: A parallelizable decentralized privacy-preserving location proximity protocol. In *2015 13th Annual Conference on Privacy, Security and Trust (PST)*, pages 1–6, July 2015.

[15] P. Hallgren, C. Orlandi, and A. Sabelfeld. PrivatePool: Privacy-Preserving Ridesharing. In *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*, pages 276–291, Aug 2017.

[16] A. Hern. Uber employees 'spied on ex-partners, politicians and Beyoncé'. 2016. https://www.theguardian.com/technology/2016/dec/13/uber-employees-spying-ex-partners-politicians-beyonce.

[17] Y. Huang, D. Evans, and J. Katz. Private set intersection: Are garbled circuits better than custom protocols? In *NDSS*. The Internet Society, 2012.

[18] Google Inc. Google Sheets - create and edit spreadsheets online, for free. https://docs.google.com/spreadsheets.

[19] Lyft Inc. Lyft: A ride when you need one | Be a Lyft driver - Lyft. https://www.lyft.com/.

[20] Uber Technologies Inc. Uber - Get a ride near you - Earn Money by Driving | Uber. https://www.uber.com/.

[21] Investopedia. Blockchain Definition. https://www.investopedia.com/terms/b/blockchain.asp.

[22] R. Jones. The Uber scammers who take users for a (very expensive) ride. 2016. https://www.theguardian.com/money/2016/apr/22/uber-scam-hacking-account-phantom-journeys.

[23] N. Juristo and A. M. Moreno. *Basics of Software Engineering Experimentation.* Springer Publishing Company, Incorporated, 1st edition, 2010.

[24] B. Kitchenham, L. Pickard, and S. L. Pfleeger. Case studies for method and tool evaluation. *IEEE Software*, 12(4):52–62, Jul 1995.

[25] J. Kleinberg and E. Tardos. *Algorithm Design.* Pearson/Addison-Wesley, 2006.

[26] V. Kolesnikov, R. Kumaresan, M. Rosulek, and N. Trieu. Efficient Batched Oblivious PRF with Applications to Private Set Intersection. Cryptology ePrint Archive, Report 2016/799, 2016. https://eprint.iacr.org/2016/799.

[27] D. Lee. Uber concealed huge data breach. 2017. http://www.bbc.com/news/technology-42075306.

[28] S. Levin. Facebook fires engineer accused of stalking, possibly by abusing data access. 2018. https://www.theguardian.com/technology/2018/may/02/facebook-engineer-fired-alleged-stalker-tinder.

[29] S. Mascetti, D. Freni, C. Bettini, X. S. Wang, and S. Jajodia. Privacy in geo-social networks: proximity notification with untrusted service providers and curious buddies. *The VLDB Journal*, 20(4):541–566, Aug 2011.

[30] J. McKay and P. Marshall. Shaping a Process Model for Action Research. *PACIS 2001 Poceedings*, 37, 2001.

[31] A. Narayanan, N. Thiagarajan, M. Lakhani, M. Hamburg, and D. Boneh. Location Privacy via Private Proximity Testing. In *In NDSS*, 01 2011.

[32] J. Nelson. Blockchains for Distributed Systems. 2016. `https://medium.com/@judecnelson/blockchains-for-distributed-systems-ffd68e6341b5`.

[33] The City of New York. NYC Taxi & Limousine Commission. `http://www.nyc.gov/html/tlc/html/home/home.shtml`.

[34] Massachusetts Institute of Technology. Kerberos: The Network Authentication Protocol. `https://web.mit.edu/kerberos/`.

[35] C. Orlandi. Is Multiparty Computation Any Good In Practice? In *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5848–5851, May 2011.

[36] P. Paillier. Public-Key Cryptosystems Based on Composite Degree Residuosity Classes. In Jacques Stern, editor, *Advances in Cryptology — EUROCRYPT '99*, pages 223–238, Berlin, Heidelberg, 1999. Springer Berlin Heidelberg.

[37] B. Pinkas, T. Schneider, G. Segev, and M. Zohner. Phasing: Private Set Intersection using Permutation-based Hashing. Cryptology ePrint Archive, Report 2015/634, 2015. `https://eprint.iacr.org/2015/634`.

[38] B. Pinkas, T. Schneider, C. Weinert, and U. Wieder. Efficient circuit-based psi via cuckoo hashing. pages 125–157, 01 2018. `https://eprint.iacr.org/2018/120`.

[39] I. Polakis, G. Argyros, T. Petsios, S. Sivakorn, and A. D. Keromytis. Where's Wally?: Precise User Discovery Attacks in Location Proximity Services. In *Proceedings of the 22Nd ACM SIGSAC Conference on Computer and Communications Security*, CCS '15, pages 817–828, New York, NY, USA, 2015. ACM.

[40] P. Runeson and M. Höst. Guidelines for conducting and reporting case study research in software engineering. *Empirical Software Engineering*, 14(2):131, Dec 2008.

[41] A. Shamir. How to share a secret. *Commun. ACM*, 22(11):612–613, November 1979.

[42] L. Šikšnys, J. R. Thomsen, S. Šaltenis, Man L. Yiu, and O. Andersen. A Location Privacy Aware Friend Locator. In N. Mamoulis, T. Seidl, T. B. Pedersen, K. Torp, and I. Assent, editors, *Advances in Spatial and Temporal Databases*, pages 405–410, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[43] Reuters Staff. EU lawmaker says Tinder breaches data protection rules. 2016. https://www.reuters.com/article/us-match-group-tinder-eu-data-protection-idUSKCN10E1RR?feedType=RSS&feedName=internetNews.

[44] Google Statisticians and Engineers. Statistics for Google Sheets. https://sites.google.com/site/statisticsforspreadsheets/.

[45] N. Trieu. Github - osu-crypto/BaRK-OPRF: Efficient Batched Oblivious PRF with Applications to Private Set Intersection (CCS 2016). https://github.com/osu-crypto/BaRK-OPRF.

[46] V. Vaishnavi, W. Kuechler, and S. (Eds.) Petter. Design Science Research in Information Systems. Technical report, 2004/17. http://desrist.org/desrist/content/design-science-research-in-information-systems.pdf.

[47] J. Šeděnka and P. Gasti. Privacy-preserving Distance Computation and Proximity Testing on Earth, Done Right. In *Proceedings of the 9th ACM Symposium on Information, Computer and Communications Security*, ASIA CCS ’14, pages 99–110, New York, NY, USA, 2014. ACM.

[48] Z. Whittaker. OneLogin security chief reveals new details of data breach. https://www.zdnet.com/article/onelogin-hit-by-data-breached-exposing-sensitive-customer-data/.

[49] Microsoft Windows. Description of Symmetric and Asymmetric Encryption. https://support.microsoft.com/en-us/help/246071/description-of-symmetric-and-asymmetric-encryption.

[50] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén. Experimentation in Software Engineering. *Springer-Verlag Berlin Heidelberg*, 1, 2012.

[51] A. C. Yao. Protocols for secure computations. In *Proceedings of the 23rd Annual Symposium on Foundations of Computer Science*, SFCS ’82, pages 160–164, Washington, DC, USA, 1982. IEEE Computer Society.

[52] A. C. Yao. How to generate and exchange secrets. In *27th Annual Symposium on Foundations of Computer Science (sfcs 1986)*, pages 162–167, Oct 1986.

[53] O. Zaleski and A. Tarter. Uber Is Now the Most Popular Taxi App in 108 Countries, Data Show. https://www.bloomberg.com/news/articles/2016-08-23/uber-is-the-most-popular-ride-hailing-app-in-108-countries.

[54] G. Zhong, I. Goldberg, and U. Hengartner. Louis, Lester and Pierre: Three Protocols for Location Privacy. In N. Borisov and P. Golle, editors, *Privacy Enhancing Technologies*, pages 62–76, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

Appendices

A

Efficiency Results (T-KEM vs. Altered BaRK-OPRF)

		Dataset sizes							
		32	**64**	**128**	**256**	**512**	**1024**	**2048**	**4096**
Protocol	**T-KEM**	0.035962	0.202503	1.394349	10.365839	79.646090	619.613000	4897.286800	38829.0150
		0.032389	0.201888	1.390940	10.280580	78.995720	618.534600	4892.787300	38827.4810
		0.032644	0.201616	1.390088	10.277747	78.998090	618.419300	4891.913900	38823.6920
		0.032551	0.202045	1.390780	10.282961	79.001580	618.490900	4892.681800	38823.9400
		0.032598	0.201728	1.390011	10.281397	78.986390	618.538000	4892.829800	38826.9120
		0.032670	0.202148	1.389531	10.276165	78.998100	618.474100	4892.362000	38827.1370
		0.032487	0.201775	1.390250	10.287111	79.015190	618.769400	4892.853500	38827.4580
		0.032676	0.201692	1.389943	10.279575	79.026600	618.746500	4893.816700	38834.0220
		0.032535	0.201807	1.389680	10.285779	79.009670	618.621100	4893.577700	38824.8820
		0.032755	0.202005	1.390871	10.277218	79.002650	618.539600	4892.336500	38828.4870
	Altered-BaRK	0.089000	0.087000	0.087000	0.087000	0.087000	0.091000	0.095000	0.099000
		0.087000	0.086000	0.084000	0.098000	0.089000	0.092000	0.093000	0.096000
		0.088000	0.086000	0.086000	0.088000	0.089000	0.092000	0.095000	0.100000
		0.087000	0.087000	0.087000	0.103000	0.090000	0.091000	0.094000	0.098000
		0.088000	0.097000	0.087000	0.090000	0.090000	0.093000	0.094000	0.097000
		0.086000	0.085000	0.088000	0.089000	0.088000	0.090000	0.105000	0.098000
		0.087000	0.085000	0.086000	0.091000	0.088000	0.093000	0.097000	0.099000
		0.086000	0.086000	0.088000	0.089000	0.099000	0.091000	0.092000	0.102000
		0.086000	0.085000	0.087000	0.088000	0.090000	0.089000	0.095000	0.099000
		0.085000	0.087000	0.087000	0.088000	0.088000	0.091000	0.095000	0.098000

B

Endpoint Method Efficiency (Without Time)

Measurement #	**First endpoint computed**	**Second endpoint computed**	**Total**
1	0.1902	0.1962	0.3864
2	0.1908	0.1979	0.3887
3	0.1942	0.1965	0.3908
4	0.1845	0.1836	0.3681
5	0.2028	0.1853	0.3881
6	0.1952	0.1901	0.3853
7	0.1921	0.1904	0.3825
8	0.1891	0.1888	0.3778
9	0.1844	0.1876	0.3720
10	0.1875	0.1937	0.3812
11	0.1977	0.2010	0.3987
12	0.1872	0.1901	0.3774
13	0.1981	0.1958	0.3939
14	0.1882	0.1907	0.3789
15	0.1928	0.1836	0.3764
16	0.1971	0.1920	0.3891
17	0.1875	0.1936	0.3811
18	0.1885	0.1920	0.3805
19	0.1864	0.1914	0.3777
20	0.1915	0.1895	0.3810
21	0.2009	0.1932	0.3941
22	0.1982	0.1827	0.3809
23	0.1863	0.1806	0.3670
24	0.1996	0.1800	0.3796
25	0.1864	0.1951	0.3815

C

Endpoint Method Efficiency (With Time)

Measurement #	First endpoint computed	Second endpoint computed	Total
1	0.3306	0.5982	0.9288
2	0.3350	0.6105	0.9455
3	0.3211	0.5768	0.8979
4	0.3193	0.5721	0.8914
5	0.7348	0.6176	1.3524
6	0.3228	0.5550	0.8778
7	0.3243	0.5920	0.9163
8	0.3358	0.5805	0.9163
9	0.3298	0.5930	0.9228
10	0.3381	0.5938	0.9319
11	0.3209	0.6125	0.9333
12	0.3347	0.5878	0.9226
13	0.3355	0.5962	0.9317
14	0.3310	0.5788	0.9098
15	0.3323	0.5918	0.9241
16	0.3250	0.6037	0.9287
17	0.3265	0.5885	0.9149
18	0.3361	0.5755	0.9116
19	0.3304	0.6092	0.9396
20	0.3379	0.5772	0.9151
21	0.3276	0.6199	0.9474
22	0.3254	0.5805	0.9059
23	0.3248	0.5991	0.9239
24	0.3259	0.6167	0.9426
25	0.3245	0.5771	0.9016

D

Intersection Method Efficiency (With Time)

			Measurement nr.									
Data-set size	**Max Time Dev.**	**Time Dev. Prec.**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
32	0	60	0.089	0.086	0.088	0.089	0.094	0.085	0.087	0.087	0.087	0.087
32	0	45	0.102	0.099	0.086	0.096	0.086	0.087	0.087	0.086	0.087	0.091
32	0	30	0.087	0.085	0.087	0.086	0.092	0.087	0.097	0.096	0.086	0.087
32	2	60	0.088	0.086	0.088	0.104	0.086	0.086	0.085	0.085	0.088	0.087
32	2	45	0.088	0.098	0.087	0.097	0.086	0.089	0.086	0.093	0.091	0.097
32	2	30	0.087	0.088	0.088	0.087	0.087	0.095	0.086	0.096	0.088	0.088
32	4	60	0.100	0.086	0.089	0.087	0.087	0.088	0.085	0.087	0.089	0.088
32	4	45	0.089	0.089	0.098	0.088	0.100	0.088	0.100	0.087	0.088	0.086
32	4	30	0.087	0.090	0.088	0.090	0.096	0.097	0.091	0.088	0.089	0.091
32	6	60	0.088	0.093	0.099	0.092	0.089	0.089	0.088	0.088	0.089	0.090
32	6	45	0.088	0.088	0.090	0.091	0.088	0.091	0.089	0.088	0.086	0.089
32	6	30	0.091	0.088	0.088	0.091	0.087	0.089	0.089	0.086	0.090	0.089
32	8	60	0.087	0.087	0.088	0.091	0.088	0.089	0.088	0.091	0.088	0.088
32	8	45	0.088	0.089	0.100	0.089	0.088	0.089	0.089	0.095	0.087	0.087
32	8	30	0.089	0.092	0.091	0.089	0.091	0.088	0.088	0.091	0.102	0.102
32	10	60	0.090	0.088	0.101	0.087	0.093	0.090	0.089	0.087	0.091	0.091
32	10	45	0.098	0.090	0.092	0.089	0.089	0.117	0.095	0.091	0.090	0.089
32	10	30	0.104	0.090	0.090	0.092	0.090	0.092	0.090	0.090	0.089	0.090
32	12	60	0.086	0.088	0.088	0.087	0.088	0.088	0.093	0.088	0.092	0.088
32	12	45	0.092	0.087	0.090	0.091	0.091	0.090	0.102	0.088	0.102	0.089
32	12	30	0.090	0.101	0.090	0.092	0.091	0.094	0.090	0.100	0.104	0.090
64	0	60	0.089	0.087	0.085	0.085	0.086	0.087	0.087	0.088	0.088	0.087
64	0	45	0.088	0.088	0.096	0.087	0.086	0.090	0.097	0.086	0.098	0.089
64	0	30	0.085	0.089	0.088	0.105	0.087	0.089	0.103	0.090	0.087	0.088
64	2	60	0.088	0.088	0.087	0.089	0.100	0.090	0.089	0.087	0.090	0.088
64	2	45	0.102	0.088	0.091	0.090	0.098	0.088	0.088	0.094	0.088	0.107
64	2	30	0.089	0.089	0.090	0.090	0.088	0.088	0.089	0.087	0.091	0.093

			Measurement nr.									
Data-set size	**Max Time Dev.**	**Time Dev. Prec.**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
64	4	60	0.089	0.092	0.087	0.089	0.088	0.090	0.089	0.089	0.099	0.097
64	4	45	0.089	0.088	0.090	0.090	0.089	0.087	0.090	0.089	0.101	0.089
64	4	30	0.101	0.090	0.098	0.092	0.100	0.089	0.090	0.090	0.090	0.091
64	6	60	0.101	0.089	0.089	0.089	0.088	0.089	0.090	0.086	0.092	0.098
64	6	45	0.090	0.088	0.089	0.099	0.092	0.090	0.088	0.089	0.089	0.088
64	6	30	0.090	0.091	0.093	0.092	0.091	0.091	0.102	0.091	0.091	0.089
64	8	60	0.090	0.091	0.090	0.090	0.087	0.089	0.088	0.091	0.091	0.094
64	8	45	0.091	0.091	0.089	0.092	0.091	0.090	0.093	0.090	0.091	0.088
64	8	30	0.101	0.096	0.091	0.092	0.092	0.093	0.092	0.094	0.092	0.091
64	10	60	0.091	0.089	0.088	0.089	0.103	0.092	0.101	0.102	0.090	0.090
64	10	45	0.102	0.089	0.100	0.093	0.098	0.090	0.090	0.090	0.089	0.090
64	10	30	0.094	0.094	0.095	0.095	0.093	0.094	0.101	0.093	0.093	0.105
64	12	60	0.105	0.094	0.090	0.090	0.090	0.091	0.091	0.095	0.090	0.091
64	12	45	0.090	0.099	0.091	0.091	0.090	0.092	0.098	0.093	0.091	0.091
64	12	30	0.104	0.095	0.096	0.094	0.093	0.093	0.096	0.103	0.107	0.093
128	0	60	0.091	0.088	0.091	0.089	0.088	0.089	0.088	0.095	0.088	0.086
128	0	45	0.091	0.087	0.087	0.086	0.087	0.089	0.088	0.087	0.087	0.088
128	0	30	0.088	0.087	0.089	0.097	0.089	0.088	0.102	0.098	0.086	0.089
128	2	60	0.088	0.087	0.087	0.090	0.088	0.087	0.086	0.086	0.088	0.089
128	2	45	0.090	0.089	0.100	0.098	0.087	0.089	0.089	0.095	0.089	0.151
128	2	30	0.089	0.089	0.093	0.089	0.097	0.088	0.087	0.089	0.091	0.101
128	4	60	0.092	0.087	0.089	0.090	0.102	0.100	0.089	0.088	0.092	0.095
128	4	45	0.093	0.092	0.100	0.090	0.089	0.101	0.091	0.102	0.090	0.090
128	4	30	0.090	0.102	0.105	0.093	0.090	0.091	0.092	0.091	0.091	0.092
128	6	60	0.093	0.091	0.091	0.090	0.094	0.092	0.090	0.090	0.090	0.091
128	6	45	0.092	0.093	0.092	0.093	0.091	0.093	0.105	0.093	0.091	0.094
128	6	30	0.105	0.094	0.092	0.093	0.093	0.095	0.094	0.101	0.096	0.103
128	8	60	0.090	0.094	0.103	0.104	0.092	0.091	0.090	0.091	0.090	0.091
128	8	45	0.094	0.092	0.092	0.095	0.092	0.092	0.095	0.093	0.091	0.092
128	8	30	0.096	0.099	0.094	0.093	0.094	0.097	0.099	0.105	0.101	0.099
128	10	60	0.106	0.095	0.093	0.091	0.093	0.093	0.093	0.099	0.092	0.093
128	10	45	0.094	0.095	0.095	0.097	0.097	0.097	0.097	0.095	0.096	0.095
128	10	30	0.099	0.097	0.096	0.097	0.098	0.109	0.096	0.098	0.098	0.096
128	12	60	0.104	0.093	0.093	0.094	0.094	0.103	0.093	0.093	0.093	0.101
128	12	45	0.099	0.106	0.100	0.097	0.108	0.095	0.096	0.096	0.096	0.093

			Measurement nr.									
Data-set size	**Max Time Dev.**	**Time Dev. Prec.**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
128	12	30	0.101	0.112	0.097	0.100	0.101	0.099	0.097	0.101	0.102	0.100
256	0	60	0.090	0.087	0.088	0.085	0.098	0.086	0.087	0.087	0.089	0.089
256	0	45	0.088	0.090	0.088	0.088	0.087	0.089	0.085	0.088	0.088	0.090
256	0	30	0.087	0.089	0.090	0.087	0.087	0.089	0.091	0.088	0.088	0.090
256	2	60	0.090	0.090	0.101	0.091	0.093	0.089	0.089	0.090	0.091	0.090
256	2	45	0.101	0.089	0.091	0.091	0.093	0.092	0.093	0.090	0.100	0.103
256	2	30	0.094	0.093	0.096	0.093	0.104	0.092	0.093	0.090	0.093	0.095
256	4	60	0.092	0.095	0.093	0.094	0.093	0.093	0.094	0.095	0.095	0.093
256	4	45	0.093	0.105	0.092	0.095	0.095	0.107	0.095	0.098	0.095	0.094
256	4	30	0.107	0.096	0.098	0.095	0.098	0.096	0.096	0.106	0.095	0.099
256	6	60	0.096	0.097	0.096	0.111	0.093	0.095	0.095	0.096	0.095	0.096
256	6	45	0.098	0.097	0.096	0.096	0.098	0.099	0.097	0.098	0.096	0.095
256	6	30	0.099	0.101	0.113	0.099	0.099	0.102	0.099	0.100	0.099	0.098
256	8	60	0.095	0.100	0.106	0.096	0.093	0.097	0.096	0.097	0.099	0.098
256	8	45	0.099	0.100	0.097	0.110	0.101	0.103	0.101	0.100	0.099	0.100
256	8	30	0.098	0.109	0.110	0.098	0.098	0.101	0.098	0.097	0.098	0.097
256	10	60	0.099	0.097	0.098	0.099	0.099	0.100	0.100	0.097	0.099	0.099
256	10	45	0.102	0.104	0.105	0.103	0.114	0.109	0.103	0.100	0.099	0.102
256	10	30	0.102	0.101	0.100	0.101	0.099	0.100	0.113	0.102	0.101	0.101
256	12	60	0.107	0.101	0.102	0.100	0.099	0.099	0.101	0.109	0.102	0.100
256	12	45	0.111	0.101	0.098	0.097	0.097	0.101	0.101	0.100	0.099	0.101
256	12	30	0.106	0.102	0.104	0.102	0.110	0.102	0.107	0.111	0.105	0.105
512	0	60	0.091	0.089	0.089	0.090	0.089	0.088	0.088	0.090	0.089	0.088
512	0	45	0.090	0.090	0.089	0.088	0.088	0.088	0.091	0.087	0.087	0.089
512	0	30	0.097	0.087	0.090	0.088	0.911	0.091	0.091	0.090	0.088	0.092
512	2	60	0.092	0.092	0.093	0.092	0.093	0.092	0.092	0.092	0.090	0.094
512	2	45	0.096	0.093	0.093	0.103	0.094	0.096	0.105	0.094	0.101	0.091
512	2	30	0.097	0.097	0.099	0.098	0.097	0.098	0.096	0.109	0.099	0.096
512	4	60	0.097	0.094	0.111	0.101	0.101	0.099	0.096	0.096	0.098	0.098
512	4	45	0.101	0.100	0.098	0.099	0.108	0.098	0.099	0.109	0.100	0.099
512	4	30	0.098	0.098	0.100	0.102	0.098	0.099	0.108	0.100	0.099	0.100
512	6	60	0.101	0.101	0.102	0.099	0.113	0.102	0.101	0.105	0.102	0.100
512	6	45	0.100	0.115	0.100	0.099	0.108	0.100	0.099	0.099	0.113	0.099
512	6	30	0.104	0.113	0.103	0.114	0.108	0.106	0.104	0.103	0.104	0.103
512	8	60	0.097	0.098	0.098	0.101	0.102	0.098	0.097	0.097	0.097	0.099

			Measurement nr.									
Data-set size	**Max Time Dev.**	**Time Dev. Prec.**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
512	8	45	0.104	0.102	0.112	0.104	0.111	0.102	0.116	0.102	0.100	0.103
512	8	30	0.109	0.112	0.109	0.118	0.110	0.109	0.109	0.108	0.109	0.112
512	10	60	0.102	0.102	0.103	0.101	0.107	0.105	0.104	0.103	0.103	0.102
512	10	45	0.106	0.104	0.108	0.107	0.105	0.107	0.118	0.107	0.107	0.107
512	10	30	0.116	0.116	0.124	0.128	0.123	0.116	0.123	0.126	0.124	0.118
512	12	60	0.106	0.104	0.105	0.104	0.105	0.103	0.115	0.116	0.105	0.104
512	12	45	0.113	0.109	0.112	0.108	0.111	0.112	0.109	0.109	0.110	0.108
512	12	30	0.120	0.121	0.121	0.121	0.124	0.120	0.119	0.122	0.122	0.121
1024	0	60	0.091	0.090	0.090	0.090	0.100	0.090	0.090	0.094	0.102	0.092
1024	0	45	0.098	0.092	0.090	0.090	0.100	0.093	0.090	0.102	0.093	0.093
1024	0	30	0.090	0.091	0.092	0.089	0.093	0.100	0.102	0.093	0.090	0.101
1024	2	60	0.099	0.100	0.099	0.100	0.107	0.100	0.098	0.108	0.110	0.109
1024	2	45	0.114	0.100	0.104	0.101	0.102	0.101	0.102	0.100	0.113	0.101
1024	2	30	0.107	0.100	0.100	0.101	0.099	0.099	0.100	0.098	0.114	0.102
1024	4	60	0.102	0.104	0.100	0.100	0.099	0.111	0.101	0.101	0.101	0.112
1024	4	45	0.104	0.107	0.112	0.106	0.107	0.114	0.105	0.106	0.104	0.106
1024	4	30	0.110	0.109	0.120	0.111	0.126	0.111	0.111	0.112	0.110	0.112
1024	6	60	0.106	0.105	0.117	0.107	0.115	0.106	0.106	0.103	0.105	0.108
1024	6	45	0.111	0.110	0.112	0.109	0.113	0.114	0.114	0.124	0.111	0.110
1024	6	30	0.123	0.123	0.131	0.121	0.123	0.122	0.124	0.122	0.124	0.120
1024	8	60	0.113	0.113	0.109	0.112	0.108	0.110	0.122	0.111	0.111	0.114
1024	8	45	0.117	0.118	0.131	0.123	0.127	0.120	0.119	0.119	0.120	0.117
1024	8	30	0.134	0.134	0.145	0.135	0.134	0.136	0.135	0.134	0.135	0.132
1024	10	60	0.118	0.115	0.132	0.116	0.124	0.117	0.125	0.115	0.116	0.127
1024	10	45	0.134	0.125	0.126	0.126	0.136	0.134	0.126	0.125	0.124	0.144
1024	10	30	0.144	0.148	0.147	0.148	0.146	0.145	0.145	0.146	0.146	0.145
1024	12	60	0.132	0.120	0.123	0.123	0.122	0.135	0.122	0.121	0.120	0.132
1024	12	45	0.132	0.135	0.134	0.143	0.141	0.134	0.133	0.135	0.133	0.133
1024	12	30	0.157	0.156	0.166	0.167	0.164	0.159	0.159	0.157	0.168	0.157
2048	0	60	0.097	0.096	0.097	0.097	0.099	0.096	0.100	0.095	0.097	0.095
2048	0	45	0.106	0.105	0.094	0.094	0.094	0.094	0.100	0.094	0.095	0.095
2048	0	30	0.094	0.096	0.095	0.095	0.104	0.096	0.097	0.095	0.094	0.095
2048	2	60	0.106	0.104	0.102	0.103	0.102	0.111	0.101	0.103	0.103	0.104
2048	2	45	0.112	0.105	0.115	0.107	0.106	0.107	0.106	0.115	0.109	0.104
2048	2	30	0.115	0.114	0.119	0.115	0.116	0.112	0.114	0.114	0.112	0.114

			Measurement nr.									
Data-set size	**Max Time Dev.**	**Time Dev. Prec.**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
2048	4	60	0.116	0.113	0.113	0.124	0.113	0.114	0.116	0.114	0.127	0.114
2048	4	45	0.124	0.120	0.135	0.122	0.123	0.120	0.130	0.124	0.131	0.129
2048	4	30	0.136	0.138	0.138	0.138	0.136	0.149	0.151	0.136	0.136	0.139
2048	6	60	0.137	0.130	0.126	0.126	0.138	0.127	0.138	0.134	0.129	0.126
2048	6	45	0.138	0.146	0.136	0.139	0.135	0.149	0.138	0.136	0.137	0.148
2048	6	30	0.165	0.162	0.163	0.168	0.163	0.162	0.162	0.173	0.164	0.167
2048	8	60	0.138	0.139	0.141	0.141	0.139	0.138	0.136	0.139	0.136	0.138
2048	8	45	0.154	0.155	0.160	0.152	0.158	0.153	0.153	0.153	0.934	0.153
2048	8	30	0.188	0.189	0.192	0.188	0.186	0.187	0.187	0.187	0.190	0.187
2048	10	60	0.151	0.150	0.148	0.149	0.151	0.152	0.149	0.152	0.150	0.163
2048	10	45	0.168	0.169	0.171	0.179	0.168	0.171	0.172	0.178	0.168	0.171
2048	10	30	0.215	0.210	0.210	0.215	0.218	0.219	0.212	0.215	0.224	0.215
2048	12	60	0.177	0.172	0.163	0.162	0.162	0.161	0.163	0.161	0.161	0.160
2048	12	45	0.190	0.201	0.189	0.194	0.188	0.186	0.186	0.196	0.189	0.185
2048	12	30	0.240	0.245	0.248	0.244	0.241	0.248	0.242	0.243	0.245	0.241
4096	0	60	0.099	0.098	0.099	0.098	0.109	0.097	0.098	0.098	0.111	0.101
4096	0	45	0.097	0.097	0.102	0.098	0.098	0.099	0.094	0.098	0.098	0.110
4096	0	30	0.098	0.101	0.100	0.097	0.097	0.099	0.109	0.114	0.107	0.101
4096	2	60	0.122	0.123	0.121	0.122	0.121	0.122	0.119	0.120	0.131	0.120
4096	2	45	0.135	0.127	0.134	0.141	0.128	0.129	0.140	0.130	0.127	0.133
4096	2	30	0.144	0.147	0.156	0.145	0.153	0.161	0.146	0.143	0.145	0.156
4096	4	60	0.157	0.145	0.144	0.144	0.147	0.144	0.157	0.144	0.146	0.146
4096	4	45	0.177	0.163	0.173	0.165	0.165	0.158	0.166	0.161	0.159	0.170
4096	4	30	0.201	0.192	0.193	0.194	0.191	0.197	0.195	0.192	0.194	0.193
4096	6	60	0.177	0.172	0.174	0.174	0.178	0.184	0.174	0.918	0.175	0.932
4096	6	45	0.198	0.195	0.195	0.194	0.194	0.204	0.203	0.193	0.193	0.200
4096	6	30	0.256	0.247	0.248	0.249	0.247	0.251	0.251	0.249	0.248	0.250
4096	8	60	0.199	0.197	0.200	0.212	0.199	0.193	0.197	0.201	0.193	0.207
4096	8	45	0.232	0.231	0.230	0.229	0.234	0.235	0.231	0.231	0.229	0.250
4096	8	30	0.289	0.283	0.284	0.284	0.294	0.286	0.286	0.282	0.288	0.284
4096	10	60	0.221	0.224	0.221	0.219	0.220	0.228	0.219	0.219	0.221	0.218
4096	10	45	0.266	0.267	0.273	0.270	0.268	0.266	0.267	0.267	0.270	0.282
4096	10	30	0.336	0.341	0.336	0.343	0.344	0.329	0.339	0.337	0.339	0.335
4096	12	60	0.249	0.253	0.248	0.248	0.252	0.250	0.252	0.247	0.250	0.248
4096	12	45	0.278	0.294	0.286	0.281	0.290	0.283	0.283	0.280	0.278	0.281

			Measurement nr.									
Data-set size	**Max Time Dev.**	**Time Dev. Prec.**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
4096	12	30	0.398	0.399	0.389	0.391	0.395	0.395	0.389	0.388	0.383	0.385