Application of Secure Two-Party Computation in a Privacy-Preserving Android App

Marco De Vincenzi
Fabio Martinelli
Ilaria Matteucci
name.surname@iit.cnr.it
IIT, Consiglio Nazionale delle Ricerche
Pisa, Italy

Stefano Sebastio
stefano.sebastio@collins.com
Collins Aerospace - Applied Research & Technology
Cork, Ireland

ABSTRACT

Data privacy has become increasingly important in recent years, with the rise of cyber threats and the unauthorized sharing of sensitive information. Our work within the E-Corridor project has focused on developing a secure framework for sharing information in multimodal transport systems, while ensuring data privacy is maintained. Our implementation of a two-party computation schema using Yao's garbled circuits in an Android mobile setting has enabled us to create an application that allows users to find points of interest, e.g., restaurants or hotels, near specific areas without sharing any personal information. The application matches user requests using the secure two-party without disclosing any of the user's preferences with external actors. We design a threat model based on LINDDUN to show the reliability of our project. It highlights also the potential of using secure computing techniques to enable information sharing while maintaining privacy. Our work demonstrates the importance of prioritizing data privacy in our increasingly interconnected world and the potential of secure two-party computing techniques in achieving this goal. Besides, this framework is flexible and can be extended to various domains where data privacy is of utmost importance.

KEYWORDS

Privacy, secure two-party, mobile app, Android, Yao.

ACM Reference Format:
Marco De Vincenzi, Fabio Martinelli, Ilaria Matteucci, and Stefano Sebastio. 2023. Application of Secure Two-Party Computation in a Privacy-Preserving Android App. In *The 18th International Conference on Availability, Reliability and Security (ARES 2023), August 29–September 01, 2023, Benevento, Italy.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3600160.3604996

1 INTRODUCTION

In recent years, data protection has become a critical topic in our society. It is not just a computer-related issue, but it also ensures human dignity, safety, and self-determination [29]. A breach of privacy can harm not only individuals but also economic and commercial interests. Data breaches, for example, have been increasing rapidly. These are incidents where information is stolen without the knowledge or authorization of the data owner [38]. IBM in its annual report shows that the global average cost for a company of a data breach increased from 4.24$ million in 2021 to 4.35$ million in 2022 [16] and it has continuously increased in the last years. In 2022, several attacks have been performed. For instance, Cash App, the most popular finance application in the App Store, was a victim of a data breach affecting approximately the data of 8.2 million users [13]. In January 2022, the International Committee of the Red Cross was attacked and data of about 500,000 people had been stolen [17]. In this scenario, the European project *E-Corridor* [4] aims to develop solutions for passengers in a multimodal transport system. In particular, we inherit from the project the running example dealing with the creation of a full privacy-preserving environment within an airport or train station area.

To address this task, we had to find cryptographic protocols which can be efficient and preserve the users' privacy also in a mobile environment. Indeed, mobile travel applications are growing their importance due to the offered flexibility and simplified transactions that contribute to a better passenger experience. For this reason, we decide to implement a privacy-preserving solution like secure two-party computation (S2PC) that is a cryptographic protocol enabling two parties to jointly compute a function while preserving the privacy of their individual inputs. It ensures that neither party learns any information about the other's input, except what can be inferred from the output. This result can be achieved through techniques such as secure multiparty computation, homomorphic encryption, or garbled circuits. The parties collaboratively compute the desired result without revealing sensitive data, maintaining confidentiality in the process. In our work, we decided to apply Yao's garbled circuits [39] integrated into an Android application, which can retrieve the restaurants or hotels matching the user's preferences. S2PC Yao's garbled circuits protocol (GCP) transforms any data analytics into a function that can be evaluated securely and preserve confidentiality by modeling it as a Boolean circuit. Then, it can mask inputs and outputs in such a way that the parties executing the function can not retrieve any information about the inputs or intermediate values of the function. The protocol preserves privacy as long as both parties do not deviate or are colluding with the attackers [37].

Our app, related to the hotels, which we call "*My Hotel*", can be installed on any Android device with a minimum API level of 10, which corresponds to Android Gingerbread 2.3.3 (2011). The

ARES 2023, August 29–September 01, 2023, Benevento, Italy

ACM ISBN 979-8-4007-0772-8/23/08…$15.00
https://doi.org/10.1145/3600160.3604996

application has an intuitive layout and it allows users to receive the matching hotels, according to their preferences, in a specific area like inside or near an airport. Regarding the hotel option, the architecture of our solution is composed of three main actors: the user's device, a service provider device called "*Totem*", which is in charge to respond to the user's request and is located in a specific area of interest like an airport, a train station, or a square to offer the hotel search service, and a database that can retrieve the available hotels. In this work, we refer to privacy using the definition provided by the U.S. National Institute of Standards and Technologies (NIST) where "*privacy is the assurance that the confidentiality of, and access to, certain information about an entity is protected.*" [28]. Hence, in our app, our main target is to assure as property the confidentiality of user's preferences, which can be reached using Yao's garbled circuits protocol (GCP) protocol as proved by Lindell *et al.* [20].

Our work propose two main novelties. The former, as we will report in Section 2, is the application of S2PC in the mobile environment. S2PC has been applied in several contexts, however, we identify a lack of contribution for the mobile, and specifically, GCP for the Android environment, where only a few and dated contributions like [23], [3] or [22] can be retrieved, while other S2PC methods like Goldreich-Micali-Wigderson (GMW) have been more tested [7]. Consequently, our main contribution is the application of a privacy-preserving solution like S2PC GCP in a mobile scenario with an Android operating system. The result is the app *"My Hotel"*, which can be used as an example of how S2PC can be applied also in the mobile environment. The latter is that our application is developed inside the European project *E-Corridor*, which aims to develop a technological framework to unleash the power of information sharing, coupled with edge-based collaborative analytics for cyber protection [4]. Our activity addresses the needs and requirements expressed by the airport and train pilot. In particular, the goal is to enable privacy-preserving, continuous and context-aware solutions for passengers to be applied in real-life scenarios to give users full control over their data. We implement our solution with the services offered by *Amadeus*, one of the largest travel technology companies, which provided us access to APIs [1] to retrieve information about travel content and services, as well as to all resources developers may need to take full advantage of the offered APIs.

The paper is structured as follows: Section 2 describes the related work, while Section 3 contains the background about Yao's protocol. Section 4 shows the application scenario with our implementation of Yao's protocol, the architecture, the back-end and the front-end of the Android app focused on hotels. Section 5 contains the threat model based on LINDDUN method, while Section 6 contains the conclusions and possible future developments.

2 RELATED WORK

The S2PC protocol with the GCP approach has been studied and implemented in several works like [15], [18], [12], [2] and [41]. All these articles describe the protocol, propose some improvements, and tests in different environments. However, they do not address the mobile environment, forgetting the possible numerous Android users [6]. Our work is an improvement because it is: i) implemented for the Android environment (thus offering a privacy-preserving solution for "data in use", potentially to billions of users); and ii) tested in a relevant scenario following the requirements of a transportation use case part of a European project. However, concerning the mobile environments, we can find some work dedicated to the testing and implementation of S2PC, using GCP and other approaches. For instance, Huang *et al.* [14] foresees the central role smartphones can cover as a secure computing platform, identifying some limits along with potentially interesting applications. Our work answers some issues raised in this article, showing a possible implementation. Costantino *et al.* [5] implements a S2PC solution on Android smartphones, using FairPlay [21], which is a framework for secure two-party and multiparty computation that allows users to write and run secure functions. In particular, they applied this framework in the mobile environment, creating the MobileFairPlay. Concerning this article, in our work, we do not use any intermediate framework, but we implement directly the GCP protocol in the application. In another work, Demmler *et al.* [7] optimize and implement the secure computation protocol GMW on mobile phones. They focused their research on the implementation of the GMW protocol and the comparison of the speed of different S2PC protocols. This research can be considered a starting point for our work, even if they applied another S2PC protocol.

3 BACKGROUND: YAO'S PROTOCOL

Yao's protocol is a cryptographic S2PC scheme that allows two parties with their secret inputs to evaluate a function on those inputs, without revealing anything to each other. The origin was in 1986 when Yao [40] described a new tool for controlling the knowledge transfer process in cryptographic protocol design. Then, Goldreich *et al.* [11] present another approach for S2PC and multiparty computation which is a polynomial-time algorithm, called GMW, that produces a protocol to play a game that leaks no partial information, provided the majority of the players is honest. To compare the two approaches we refer to the work of Schneider *et al.* [35]: with respect to the basic GMW approach, without the implementation they suggested, the GCP can be considered less complex and more efficient to be implemented. For this reason, we decide to use Yao's GCP approach. Besides, Lindell *et al.* provides a complete description of Yao's protocol, along with a rigorous proof of security [20], which we follow to define Yao's GCP protocol security.

During the S2PC, two parties, P_1 and P_2, want to join their private inputs x and y, respectively, to compute a function $f(x, y) = (f_1(x, y), f_2(x, y))$ such that P_1 receives $f_1(x, y)$ as the first component of the output calculated by f and P_2 receives the second component $f_2(x, y)$ [20]. Firstly, in the GCP protocol, P_1 has to transform the function f into the corresponding boolean circuit C_f. The circuits are called garbled because garbling is the process where the boolean gate truth table is obfuscated [39]. Now, P_1 turns C_f into its garbled version C_g and garbles his input, x, so that it fits C_g and sends the garbled input to P_2 with the complete garbled circuit C_g [24]. Now, P_2 has the complete circuit C_g and P_1's input, but not the garbling procedure to use their input. At this step, Yao's protocol introduces another protocol: the Oblivious Transfer (OT), which is a two-party protocol between a sender and a receiver, by which the sender transfers information to the receiver. In this

case, circuits and the sender remain oblivious to what information the receiver obtains [36]. P_2 receives the garbled version of y by using OT. Then, P_2 computes the garbled circuit C_g and outputs the ungarbled result to P_2 and it concludes the protocol [24].

4 RUNNING EXAMPLE

Within the E-Corridor project, the airport and train (AT) pilot is devoted to the demonstration of solutions for multi-modal travels. Indeed, passenger journeys are usually not confined between departing and destination airports. More often, for either personal preferences or just limits in the available offer, passengers adopt multiple modes of transportation during their journey e.g., passengers could use a car sharing service to reach the closest train station, and from there they could take a train connecting to the airport. Solutions designed in the E-Corridor project span from authentication to passenger processing, access to the airport's and train station's services, to mechanisms for enhanced security and situational awareness. Albeit not critical to the transportation mission, ancillary services offered by transportation service providers represent a significant market in the order of tens of billions, expected to further grow [33]. A lack of an adequate set of services supporting passengers, while planning or during the journey, produces a fragmented trip experience. Ancillary services offered by the airlines could be categorized among products (e.g., priority boarding or extra bag), loyalty (e.g., frequent flier program) and originated by partnerships (e.g., vacation packaging). One of the service part of the latter category, it could be the need of a passenger to book a restaurant or an hotel near an airport during a journey.

Providers of global distribution system (GDS) are in charge of enabling and facilitating transactions between service providers part of the travel industry such as airlines, hotels and travel agencies. Consequently, meeting the passenger needs is important to increase the appealing of the GDS offers to service providers. In turn, the transportation service providers are heavily investing in improving the passenger experience. In light of this scenario, answering to the citizen demand for data privacy constitutes a flywheel to the business of the GDS. It is also worth considering that a common business model adopted by GDS is based on fees applied to the confirmed booking transactions. Therefore, preserving the passengers privacy during the search of a hotel does not affect negatively the current business model nor it requires to change that. Our application addresses this scenario by providing an efficient app to find restaurants or hotels matching the users preferences in a privacy-preserving way. To ensure relevance, we tested our solution in hotels using Amadeus APIs, a leading GDS provider for the global travel and tourism industry. While Amadeus provides hotel information, but not restaurants, we tested the option to include restaurants by creating a separate app version with a simulated restaurant database. The processes and solutions used were the same as the hotel app version. We will now focus on describing only the hotel app version, which integrates with a real Amadeus database. In our work, "My Hotel" is an Android application that can be installed on any device with at least Android Gingerbread 2.3.3 (2011). In our testing, we use a Samsung Galaxy J7 with Android Nougat 7.0 (2016) to deploy the application's front and a low-cost device like a Raspberry Pi 4 with LineageOS, an Android-based operating system, with installed our server-side Android application.

4.1 Architecture

Figure 1 reports the architecture of our implementation. In particular, it shows the three main actors: the user's device, the totem, and Amadeus's database. The *User's device* can be any device like a smartphone with Android OS with installed the application *"My Hotel"*. The *Totem* is a static device that represents the server in our S2PC schema and it can be located in a specific area like an airport or a square to cover with its wireless signal the area.

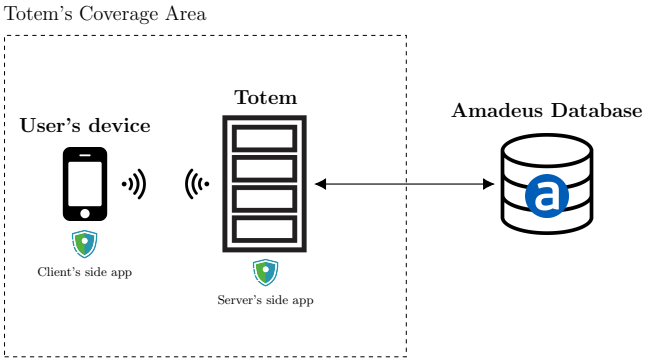

Figure 1: Architecture schema with the three main actors of our implementation.

Amadeus'server is the hotel's virtual database which can provide the list of hotels with their amenities. We create a free Amadeus profile that allows us to retrieve a limited number of hotels per second, however sufficient for our testing purposes. In particular, in Table 1, we report the called Amadeus APIs with data that the Totem shares as clear text with the external actor, in this case the Amadeus server. The Totem receives the hotels' list, convert it into circuits, and then, using the preferences, received by the user in a circuit format, it compares both circuits and it retrieves the suitable hotels.

In this architecture, when the user is inside the covered area by the totem signal, it can connect their device to the totem and require the hotel search service, or in the case of the restaurant app, the suitable restaurants. Figure 2 shows our sequence schema, with the three main actors in our implementation:

- From the device's side, the user has just to connect the personal device to the totem (calls 1-2) and send the request with the preferences (call 3). After the computation, it receives the results (call 9);
- The totem receives the request from the user and it sends two different requests to the Amadeus' server. The first, HotelList (call 4-5), retrieves the list of available hotels using the number of guests and the dates required by the user. The second, HotelSearch (call 6-7), retrieves information about amenities and services offered by only the available hotels. Then, in call 8, it computes the circuits for the hotel inputs and compares them with the users' circuits. After the computation, it sends back user's device the results;

Table 1: Amadeus APIs called by the Totem with the shared information in clear text. In italics the information directly related to the user.

API	Result	Shared information
HotelList	List of hotels in a defined area	Airport Location (latitude and longitude); default radius of search 40 Km
HotelSearch	List of available hotels with their facilities following the request	*Number of guests; check-in and check-out dates; number of rooms*

- Amadeus server has to provide the hotels' list with their facilities and prices. It communicates only with the totem (calls 4 to 7), receiving the request of the available hotels for specific dates, a defined number of guests, and in a specific area. In particular, we decide to use 40 Km as the default value, which, later, can be redefined by the totem, according to the user's circuits. Amadeus server returns the hotels' list with their characteristics It does not receive any user's preference or any information about the user's identity.

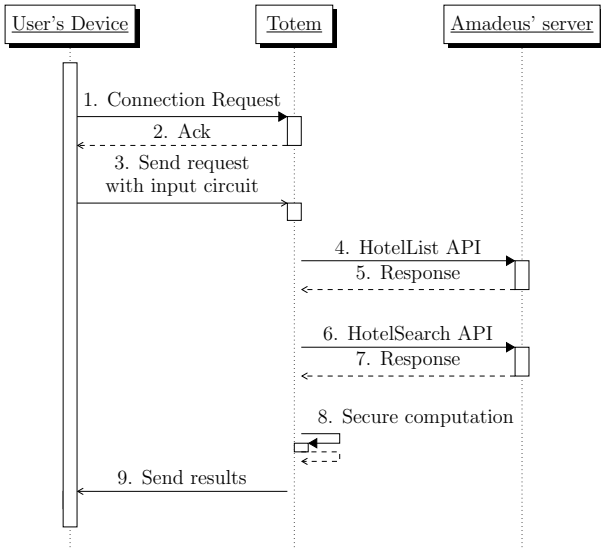

Figure 2: UML sequence diagram of a request from a user's device with the application *"My Hotel"* and the following collection of data from the Amadeus' server.

4.2 Application's Back-End

The back-end runs on the server-side Android application. We develop our Android application using the Android Studio integrated development environment (IDE) and the language Java. The core of our application is the analytic and GCP protocol. We implement a function in C language, f_{comp}, which can compare two values and retrieve the highest value. To apply Yao's protocol, we need to transform our function into garbled circuits. To perform the compilation, we use `cbmc-gc` [10], which is a circuit compiler that produces Boolean circuits from a subset of ANSI-C and constructs a protocol for securely computing f in the presence of semi-honest adversaries. In particular, we use the `cbmc-gc` version 0.9.3. We compile the circuit and `cbmc-gc` produces circuits for 32-bit integer variables and it generates several `txt` files containing the circuits, that should be saved both in the client and server. Then, using the same compiler, it is possible to give in input a numeric value and compile it in a circuit, which is saved in a file called `inputs.Client.txt` for the client's inputs and `inputs.Server.txt` for the server's inputs.

This compiler allows us to transform the comparison function into circuits and save them into our client and totem. Then, just giving the compiler the inputs, it executes the procedures as described in Section 3. In our Android apps, we use a specific module in charge of performing the computation operation on the circuits: we used a module called "CBMCAndroidlib" that is present in both the client and server sides. This module contains all the necessary Java classes for managing the Yao protocol. It includes the implementation of the Oblivious Transfer and the garbled circuit evaluation operation. To ensure flexibility and modifiability without altering the core of "CBMCAndroidlib", all other Java classes, with the graphic features, are implemented in a separate module in each app. The circuit comparison operations are executed within the totem, labeled as P_2, which then transmits the results to the client. This approach enables the creation of an app comprising two modules, with the first performing privacy-preserving operations, while the second can be adjusted to suit different requirements, such as providing services like suitable hotels or restaurants. In conclusion, the use of separate modules facilitates adaptability and customization, while preserving the core functionality of the app.

4.3 Application's Front-End

In this section, we describe the front-end, which is the presentation layer with whom the user interacts. As shown in Figure 3, the front-end is composed of two different pages. The first, on the left, is the home page where the user can press the button *Find* to start the search of the hotels, based on their previous preferences. Instead, if the user wants to change the preferences, they can press the button *Preferences*, which calls the second page on the right. On this page, the user can insert the number of guests and the check-in and check-out dates, which are the only data that can be known by the totem and the Amadeus server. Then, the user can select additional preferences that will not be disclosed to the Amadeus server. In particular, in our implementation, we just select four preferences to test our application namely distance, price, hotel category and type of bed. However, it is possible to insert different or more preferences, because the implemented Yao's protocol can support more options.

These data are converted into numeric values and then to a circuit which is sent to the totem. With this solution, the external actors are not able to know anything about the preferences, so we can have a privacy-preserving process that can be increased without affecting or modifying the back-end protocol. To begin the search for hotels located within a 40 km default radius around the airport or according to the selected preference, the user simply needs to click the *Find* button on the homepage. The system initiates the computation process and displays the results on the same page. The search results will include the names of hotels that match the user's preferences. While a booking process could be integrated, this feature was not part of our research scope. Our primary focus was on exploring the possibility of sharing preferences in a secure and privacy-preserving manner, so we left the booking process to possible future implementations.

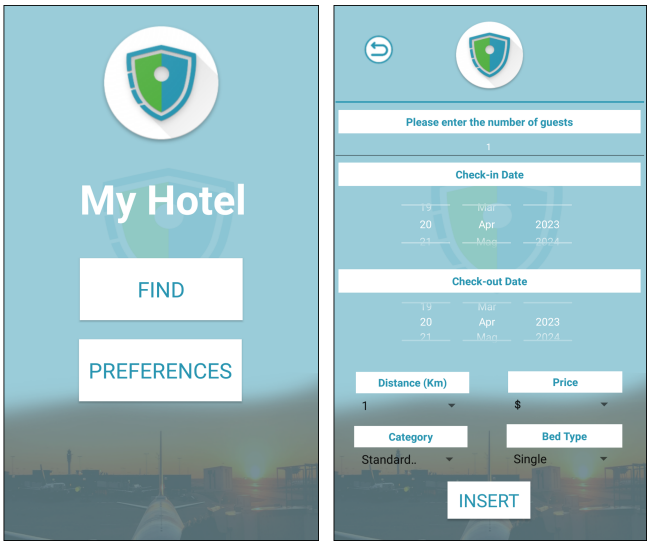

Figure 3: *My Home*' front-end pages.

5 THREAT MODEL

To test the effectiveness of the privacy solution in our scenario, we decide to create a threat model that helps us identify potential vulnerabilities and necessary mitigations. We use Threat Dragon [30], which is an open-source threat modelling tool from Open Worldwide Application Security Project (OWASP). This tool allows to design a schema with actors, processes, and indicate vulnerabilities using STRIDE [31], LINDDUN [19], or CIA [25] models.

In Figure 4, we report the three main actors of our architecture and we consider the user's device a secure environment for storing and transforming preferences into circuits for the S2PC. However, we view the totem and Amadeus' server as insecure, along with any communication between them. The totem's role is to perform the S2PC matching process between the user's circuit and Amadeus' results. In the untrusted area, we assume to face both active and passive external attackers like in the Dolev-Yao model, where they can sniff, intercept, and tamper messages and are only limited by the constraints of the applied cryptographic method [8]. While, at the same time, we consider our two actors in the untrusted area as semi-honest, so they are legitimate participant in the communication protocol and they will not deviate from the defined protocol, but attempt to learn all possible information from legitimately received messages [34]. In fact, Yao's threat model considers semi-honest adversaries, which are attackers trying to steal information, but following the protocol as specified [20]. In the case of a participant that becomes an attacker with full capacity to alter the circuits, the Yao protocol becomes vulnerable. In every message-based two-party protocol, one party learns the final output before the other. If that party is corrupt and malicious, it can refuse to send back the result to the honest party that will not learn the output [9]. In our scenario, as stated before, we consider the totem as honest but curious, so it can not refuse to send back the result to the user's device. In case of corruption of the totem, the attacker can know the name of the matching restaurants or hotels without, however, knowing the user's preferences.

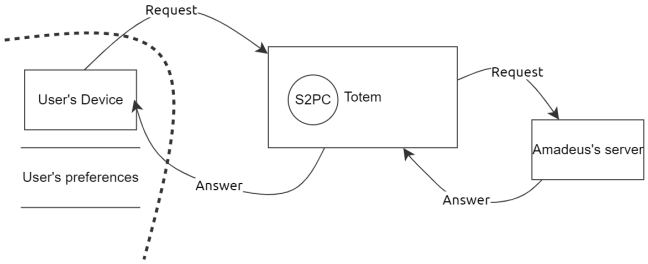

Figure 4: Threat schema designed with OWASP Threat Dragon. The dotted line represents the border of the trusted boundary, while the circle is the S2PC process.

As threat schema, we decide to apply LINDDUN model that is focused on privacy, while the other two models, STRIDE and CIA, are more related to security. LINDDUN acronym stands for Linkability, Identifiability, Non-repudiation, Detectability, Disclosure of information, Unawareness, and Non compliance, and we use it to identify potential threats and vulnerabilities:

- Linkability: it is a deduction that can be made by the attackers based on the inference rules [34]. The user's device and the totem are connected wirelessly, which means that the communication channel between them can be intercepted and monitored by attackers to link two items of interest without knowing the identity of the data subject(s) involved. However, even attackers intercept the communication, they can retrieve only the user's circuits, which are useless and unlinkable without additional information. Regarding the honest but curious participants, Amadeus's server does not know anything about the user, but it receives only a request for specific dates, while the totem knows only the results which can be used to infer some information about the preferences;
- Identifiability: it is the property where information can be used to distinguish or trace an individual's identity [26]. In our scenario, the user's device and the totem can be identified through their unique IP addresses or MAC addresses, which

can be used to track the user's movements and behavior. However, personal information, such as user identification number and preferences, are not transmitted in clear text. Instead, they can be encrypted and transmitted through circuits;
- Non-repudiation: It is the assurance that the sender is provided with proof of delivery and the recipient is provided with proof of the sender's identity [27]. The totem and Amadeus' server should have a system in place to ensure that messages are not tampered with or forged during transmission by unauthorized actors. For this reason, for each request, the identity of the actors is verified using authentication tokens. Besides, both participants as honest but curious can just observe information and not refuse to follow the protocol;
- Detectability: it is the state to be detectable so the system should prevent that an adversary is able to distinguish whether an item of interest about a data subject exists or not, regardless of being able to read the contents itself [19]. For this reason, we use the Yao's circuits which can not provide any information to distinguish whether an item of interest about a data subject exists or not, even if intercepted;
- Disclosure of information: user's information should be protected against possible disclosures. For this reason, out-of-the trusted area, information are encrypted or transmitted with Yao's circuits;
- Unawareness: the system should ensure that users are aware on how sensitive information, such as the user's preferences and personal data, are treated [19]. The E-Corridor project ensures with the Data Sharing Agreements (DSAs) that all users are aware on how and by who their information can be used;
- Non compliance: the system should assure that the processing, storage, or handling of personal data is compliant with legislation, regulation, and/or policy [19]. Our solution is compliant with the European General Data Protection Regulation (GDPR) [32], which is legally binding in the European countries where the E-Corridor project was developed. In particular, with the DSAs we assure a complete control of the users on their data and the protection using privacy-preserving solutions like Yao's protocol.

To conclude, during the designing phase, the LINDDUN model helped us identify potential threats and vulnerabilities in our scenario. As reported in this section, we applied mitigations and solutions to ensure that the system is privacy-preserving against active or passive attackers with semi-honest actors. In particular, in our scenario, the Yao schema provides us all the desired properties like confidentiality to assure the privacy of user's preferences.

6 CONCLUSION

Data privacy is a growing concern since when the European Union's General Data Protection Regulation (GDPR) [32] went into effect in May 2018, there has been a constant increase in such a kind of regulations across the globe that are getting always stricter. The transportation sector is one of the largest industries, and service providers are working to improve the passenger experience. Following the passenger needs and service access preferences is thus of primary importance for their market. Therefore, requests for smartphone access and increased privacy are two leading drivers for any new service. To address the requirements of the airport and train scenarios within the European E-Corridor project, we developed an Android application for mobile devices that implements Yao's protocol for secure two-party computation. Specifically, we created an application that retrieves a list of available hotels based on the user's preferences, while the GCP protocol ensures the privacy of these preferences even in a mobile environment. This solution enables us to find hotels that match the user's preferences within a specific range, without revealing any information about the preferences to the service provider. Our implementation can be adapted for other data analysis functions and can operate in a mobile environment while preserving input data privacy. In particular, with the two-module structures, the Java modules can be used as a basic skeleton to implement different services.

In conclusion, our work proposes a privacy-preserving solution to enable users to receive service according to their preferences in a specific area. In future work, we plan to study solutions in case of malicious actors, and not only semi-honest participants, and to explore other privacy-preserving solutions like differential privacy to compare different S2PC schemes such as Yao and GMW. We will evaluate the pros and cons of each solution, taking into account also timing and computational resources.

ACKNOWLEDGMENTS

The project leading to this application has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement No 883135 (E-Corridor). This work was partially supported by project SERICS (PE00000014) under the NRRP MUR program funded by the EU - NGEU.

REFERENCES

[1] Amadeus. 2023. Amadeus for developers. https://developers.amadeus.com/ Accessed on February 2, 2023.
[2] Osman Biçer. 2017. Efficiency Optimizations on Yao's Garbled Circuits and Their Practical Applications. *CoRR* abs/1703.03473 (2017). arXiv:1703.03473 http://arxiv.org/abs/1703.03473
[3] Henry Carter, Charles Lever, and Patrick Traynor. 2014. Whitewash: outsourcing garbled circuit generation for mobile devices. In *Proceedings of the 30th Annual Computer Security Applications Conference, ACSAC 2014, New Orleans, LA, USA, December 8-12, 2014*, Charles N. Payne Jr., Adam Hahn, Kevin R. B. Butler, and Micah Sherr (Eds.). ACM, 266–275. https://doi.org/10.1145/2664243.2664255
[4] Consiglio Nazionale delle Ricerche. [n. d.]. E-Corridor. https://e-corridor.eu/ Accessed on October 28, 2022.
[5] Gianpiero Costantino, Fabio Martinelli, Paolo Santi, and Dario Amoruso. 2012. An Implementation of Secure Two-Party Computation for Smartphones with Application to Privacy-Preserving Interest-Cast. In *Proceedings of the 18th Annual International Conference on Mobile Computing and Networking* (Istanbul, Turkey) *(Mobicom '12)*. Association for Computing Machinery, New York, NY, USA, 447–450. https://doi.org/10.1145/2348543.2348607
[6] David Curry. 2023. Android Statistics (2023). https://www.businessofapps.com/data/android-statistics/ Accessed on January 21, 2023.
[7] Daniel Demmler, Thomas Schneider, and Michael Zohner. 2014. Ad-Hoc Secure Two-Party Computation on Mobile Devices Using Hardware Tokens. In *Proceedings of the 23rd USENIX Conference on Security Symposium* (San Diego, CA) *(SEC'14)*. USENIX Association, USA, 893–908.
[8] D. Dolev and A. Yao. 1983. On the security of public key protocols. *IEEE Transactions on Information Theory* 29, 2 (1983), 198–208. https://doi.org/10.1109/TIT.1983.1056650
[9] David Evans, Vladimir Kolesnikov, and Mike Rosulek. 2018. .

[10] Github. 2020. *CBMC-GC*. https://github.com/MPC-SoK/frameworks/blob/master/cbmc-gc/README.md Accessed on January 21, 2023.
[11] O. Goldreich, S. Micali, and A. Wigderson. 1987. How to Play ANY Mental Game. In *Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing* (New York, New York, USA) *(STOC '87)*. Association for Computing Machinery, New York, NY, USA, 218–229. https://doi.org/10.1145/28395.28420
[12] S. Dov Gordon, Jonathan Katz, Vladimir Kolesnikov, Fernando Krell, Tal Malkin, Mariana Raykova, and Yevgeniy Vahlis. 2012. Secure Two-Party Computation in Sublinear (Amortized) Time. In *Proceedings of the 2012 ACM Conference on Computer and Communications Security* (Raleigh, North Carolina, USA) *(CCS '12)*. Association for Computing Machinery, New York, NY, USA, 513–524. https://doi.org/10.1145/2382196.2382251
[13] Maria Henriquez. 2022. Block confirms Cash app breach affecting 8m users. https://www.securitymagazine.com/articles/97396-block-confirms-cash-app-breach-affecting-8m-users Accessed on January 17, 2023.
[14] Yan Huang, Peter Chapman, and David Evans. 2011. Privacy-preserving applications on smartphones, Vol. 8. 4–4.
[15] Yan Huang, David Evans, Jonathan Katz, and Lior Malka. 2011. Faster Secure Two-Party Computation Using Garbled Circuits. In *Proceedings of the 20th USENIX Conference on Security* (San Francisco, CA) *(SEC'11)*. USENIX Association, USA, 35.
[16] IBM. 2022. IBM - Cost of a data breach 2022. https://www.ibm.com/reports/data-breach?utm_content=SRCWW&p1=Search&p4=43700072664611161&p5=e&gclsrc=ds Accessed on January 17, 2023.
[17] ICRC. 2022. Cyber attack on ICRC: What we know. https://www.icrc.org/en/document/cyber-attack-icrc-what-we-know Accessed on January 17, 2023.
[18] Stanisław Jarecki and Vitaly Shmatikov. 2007. Efficient Two-Party Secure Computation on Committed Inputs. In *Advances in Cryptology - EUROCRYPT 2007*, Moni Naor (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 97–114.
[19] KU Leuven. 2023. LINDDUN. https://www.linddun.org/ Accessed on April 2, 2023.
[20] Yehuda Lindell and Benny Pinkas. 2009. A Proof of Security of Yao's Protocol for Two-Party Computation. *Journal of Cryptology* 22, 2 (01 Apr 2009), 161–188. https://doi.org/10.1007/s00145-008-9036-8
[21] Dahlia Malkhi, Noam Nisan, Benny Pinkas, and Yaron Sella. 2004. Fairplay—a Secure Two-Party Computation System. In *Proceedings of the 13th Conference on USENIX Security Symposium - Volume 13* (San Diego, CA) *(SSYM'04)*. USENIX Association, USA, 20.
[22] Benjamin Mood, Debayan Gupta, Kevin R. B. Butler, and Joan Feigenbaum. 2015. Reuse It Or Lose It: More Efficient Secure Computation Through Reuse of Encrypted Values. *CoRR* abs/1506.02954 (2015). arXiv:1506.02954 http://arxiv.org/abs/1506.02954
[23] Benjamin Mood, Lara Letaw, and Kevin R. B. Butler. 2012. Memory-Efficient Garbled Circuit Generation for Mobile Devices. In *Financial Cryptography and Data Security - 16th International Conference, FC 2012, Kralendijk, Bonaire, Februray 27-March 2, 2012, Revised Selected Papers (Lecture Notes in Computer Science, Vol. 7397)*, Angelos D. Keromytis (Ed.). Springer, 254–268. https://doi.org/10.1007/978-3-642-32946-3_19
[24] Frederic Naumann. 2016. Garbled circuits. In *Seminar Innovative Internet-Technologien und Mobilkommunikation SS2016*.
[25] NIST. 2023. CIA. https://www.nccoe.nist.gov/publication/1800-25/VolA/index.html Accessed on April 2, 2023.
[26] NIST. 2023. NIST Identifiability definition. https://csrc.nist.gov/glossary/term/personally_identifiable_information Accessed on April 14, 2023.
[27] NIST. 2023. NIST Non-Repudiation definition. https://csrc.nist.gov/glossary/term/non_repudiation#:~:text=NIST%20SP%20800%2D57%20Part,deny%20having%20process%20the%20information%20. Accessed on April 14, 2023.
[28] NIST. 2023. NIST Privacy definition. https://csrc.nist.gov/glossary/term/privacy#:~:text=Definitions%3A,from%20NIST%20SP%20800%2D130 Accessed on January 21, 2023.
[29] OVIC. 2023. The Importance of Privacy. https://ovic.vic.gov.au/privacy/resources-for-organisations/privacy-officer-toolkit/the-importance-of-privacy/#:~:text=Human%20right%20to%20privacy&text=It%20relates%20to%20an%20individual's,freely%20develop%20their%20own%20personality Accessed on January 13, 2023.
[30] OWASP. 2023. OWASP Threat Dragon. https://owasp.org/www-project-threat-dragon/ Accessed on April 2, 2023.
[31] OWASP. 2023. STRIDE model. https://owasp.org/www-community/Threat_Modeling_Process Accessed on April 2, 2023.
[32] European Parliament and Council of the European Union. 2016. EU General Data Protection Regulation (GDPR): Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation), OJ 2016 L 119/1. https://eur-lex.europa.eu/eli/reg/2016/679/oj
[33] The Insight partners. 2019. Airline Ancillary Services Market to 2027 - Global Analysis and Forecasts by Type and Carrier Type. https://www.theinsightpartners.com/reports/airline-ancillary-services-market Accessed on February 2, 2023.
[34] Andrew J. Paverd and Andrew C. Martin. 2014. Modelling and Automatically Analysing Privacy Properties for Honest-but-Curious Adversaries.
[35] Thomas Schneider and Michael Zohner. 2013. GMW vs. Yao? Efficient Secure Two-Party Computation with Low Depth Circuits. In *Financial Cryptography and Data Security*, Ahmad-Reza Sadeghi (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 275–292.
[36] Berry Schoenmakers. 2011. *Oblivious Transfer*. Springer US, Boston, MA, 884–885. https://doi.org/10.1007/978-1-4419-5906-5_9
[37] Peter Snyder. 2014. Yao's Garbled Circuits: Recent Directions and Implementations. In *Literature review, Dept. of Computer Science, University of Illinois at Chicago*.
[38] Trend. 2023. Data Breach definition - Trend Micro. https://www.trendmicro.com/vinfo/us/security/definition/data-breach Accessed on January 17, 2023.
[39] Sophia Yakoubov. 2017. A Gentle Introduction to Yao's Garbled Circuits. (2017). https://doi.org/sonka89/www/papers/2017ygc.pdf
[40] Andrew Chi-Chih Yao. 1986. How to generate and exchange secrets. In *27th Annual Symposium on Foundations of Computer Science (sfcs 1986)*. 162–167. https://doi.org/10.1109/SFCS.1986.25
[41] Jing Zhang, Yongli Tang, Shoushan Luo, Yixian Yang, and Yang Xin. 2019. Secure two-party computation of solid triangle area and tetrahedral volume based on cloud platform. *PLOS ONE* 14 (06 2019), e0217067. https://doi.org/10.1371/journal.pone.0217067