5-2014

Privacy protection in context aware systems.

Anala Aniruddha Pandit
University of Louisville

PRIVACY PROTECTION IN CONTEXT AWARE SYSTEMS

By

ANALA ANIRUDDHA PANDIT
M.Sc. (PHYS)
M.S. (EE)
M.S. (CS)

A Dissertation
Submitted to the Faculty of the
J.B.Speed School of Engineering of the University of Louisville
in Partial Fulfillment of the Requirements
for the Degree of

Doctor of Philosophy

Department of Computer Engineering and Computer Science
University of Louisville
Louisville, Kentucky

May 2014

PRIVACY PROTECTION IN CONTEXT AWARE SYSTEMS

By

ANALA ANIRUDDHA PANDIT
M.Sc. (PHYS)
M.S. (EE)
M.S. (CS)

A Dissertation Approved

on March 4th 2014

by the following Dissertation Committee:

Dr. Anup Kumar - Dissertation Director

Dr. Adel Elmaghraby

Dr. A. Desoky

Dr. R. Ragade

Dr. S. Heragu

Dr. S.Y. Wu

DEDICATION

DEDICATED TO MY FATHER IN LAW (LATE PROF. B. R. PANDIT) AND MY PARENTS
(SHRI DEVIDAS AND MRS SUMEDHA KULKARNI)

ACKNOWLEDGMENTS

I wish to express my deep gratitude to Dr. Anup Kumar, whose constant guidance lead me towards my goal gently but firmly. I thank him for his faith and belief in my ability which gave me courage to explore. I wish to thank Dr. S. Heragu for the financial support that has opened new areas of work in designing the web application while enabling me to have a stable livelihood. I wish to thank all the faculty of CECS department who have been patient with my learning curve and always helped me tide over my personal hesitations, especially Dr. Desoky without whose help, I would not have reached this far. I wish to thank Dr. Wu, who has been my source of inspiration from 1983, when I first joined his class. Thank you for being a constant in my life. I also wish to thank Dr. Elmaghraby and Dr. Ragade for their detailed suggestions, support in all my endeavors and agreeing to be a part of my committee. I wish to express my sincere gratitude to Dr. Sam Bell (Jr.) who kept my desire to complete Ph.D. alive in me for 25 years. I would like to take this opportunity to sincerely thank Dr. Ashok Agrawala of University of Maryland for his constant support and encouragement. I appreciate the help given by my colleague Mr. Phani Polina time to time since I have joined this department. I would like to sincerely thank Dr. B. D. Kulkarni of NCL, Pune and Dr. Sujit Jogwar of ICT, Mumbai for their valuable suggestions.

I would like to mention the pillars of support in my life, without whom, it would not have been possible to reach the goal. I cannot thank my husband Dr. Aniruddha Pandit enough, to whose opinion I am addicted - for reading, re-reading, and helping me edit every single draft of this document and all other work I have done. My daughter Sphoorti, keeps pushing me to excel; my late father in law Prof. B.R. Pandit, my parents Shri Devidas and Mrs. Sumedha Kulkarni and the entire family, have always said “We are with you” and have been with me through all my joys and tears. My students have been a source of constant inspiration and they almost compel me to excel. Last but not the least, I would like to thank the Board of Governors, the

Director and colleagues of “Veermata Jijabai Technological Institute”, Mumbai, who granted me leave to complete my Ph.D.

ABSTRACT

PRIVACY PROTECTION IN CONTEXT AWARE SYSTEMS

ANALA ANIRUDDHA PANDIT

MARCH 4TH 2014

Smartphones, loaded with users' personal information, are a primary computing device for many. Advent of 4G networks, IPV6 and increased number of subscribers to these has triggered a host of application developers to develop softwares that are easy to install on the mobile devices. During the application download process, users accept the terms and conditions that permit revelation of private information. The free application markets are sustainable as the revenue model for most of these service providers is through profiling of users and pushing advertisements to the users. This creates a serious threat to users privacy and hence it is important that "privacy protection mechanisms" should be in place to protect the users' privacy. Most of the existing solutions falsify or modify the information in the service request and starve the developers of their revenue.

In this dissertation, we attempt to bridge the gap by proposing a novel integrated CLOPRO framework (Context Cloaking Privacy Protection) that achieves *Identity* privacy, *Context* privacy and *Query* privacy without depriving the service provider of sustainable revenue made from the CAPPA (Context Aware Privacy Preserving Advertising). Each service request has three parameters: identity, context and actual query. The CLOPRO framework reduces the risk of an adversary linking all of the three parameters. The main objective is to ensure that no single entity in the system has all the information about the user, the queries or the link between them, even though the user gets the desired service in a viable time frame. The proposed comprehensive framework for privacy protecting, does not require the user to use a modified OS or the service provider to modify the way an application developer designs and deploys the application and at the same time protecting the revenue model of the service provider.

The system consists of two non-colluding servers, one to process the location coordinates (Location server) and the other to process the original query (Query server). This approach makes

several inherent algorithmic and research contributions. First, we have proposed a formal definition of privacy and the attack. We identified and formalized that the privacy is protected *if* the transformation functions used are non-invertible. Second, we propose use of clustering of every component of the service request to provide anonymity to the user. We use a unique encrypted identity for every service request and a unique id for each cluster of users that ensures *Identity* privacy. We have designed a *Split Clustering Anonymization Algorithms (SCAA)* that consists of two algorithms *Location Anonymization Algorithm (LAA)* and *Query Anonymization Algorithm (QAA)*. The application of *LAA* replaces the actual location for the users in the cluster with the centroid of the location coordinates of all users in that cluster to achieve *Location* privacy. The time of initiation of the query is not a part of the message string to the service provider although it is used for identifying the timed out requests. Thus, *Context* privacy is achieved. To ensure the *Query* privacy, the generic queries (created using *QAA*) are used that cover the set of possible queries, based on the feature variations between the queries. The proposed CLOPRO framework associates the ads/coupons relevant to the generic query and the location of the users and they are sent to the user along with the result without revealing the actual user, the initiation time of query or the location and the query, of the user to the service provider.

Lastly, we introduce the use of *caching* in query processing to improve the response time in case of repetitive queries. The Query processing server caches the query result.

We have used multiple approaches to prove that privacy is preserved in CLOPRO system. We have demonstrated using the properties of the transformation functions and also using graph theoretic approaches that the user's *Identity*, *Context* and *Query* is protected against the curious but honest adversary attack, fake query and also replay attacks with the use of CLOPRO framework .

The proposed system not only provides $'k'$ anonymity, but also satisfies the $< k, s >$ and $< k, T >$ anonymity properties required for privacy protection. The complexity of our proposed algorithm is $O(n)$.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

The number of people using smart phones has been increasing exponentially with a global subscription over 1.2 billion as of March 2013 [1]. It is expected that the number of mobile devices will exceed the world population by 2014 [2]. Most people who carry a smart Internet Enabled mobile device, perform transactions or use various applications available on the smart phones (mobile banking, reservations, searches for restaurants, movies, places, check ins, Facebook, Pictures and video transmissions etc.). Most common amongst the Context based services are the Location based services. According to Rip Gerber, founder of Location – As - A – Service, application developers can double the Return on Investment (ROI) at the least, if they add location to their mobile campaign, since location based advertising increases the advertisement response rates with relevancy [1]. As a consequence of this, multitudes of applications are being developed every day for smart phones that utilize user contexts in order to be able to offer more personalized services [3]. The service providers are collecting as much information as possible from the mobile device users without their knowledge to ensure revenue.

When a person wants to use an application, (s)he downloads it even if (s)he is aware of compromising his/her privacy and the possibilities of his/her information being sent to advertisers. There is no provision to choose the permissions regarding the information the service provider will collect from your device. Also, even if the user is privacy aware, it is difficult to evade breach of privacy threats because the wide range of services needs the static (age, educational level, preferences etc.) / dynamic (Current Temperature, person nearby, various sensor readings etc.) context to provide these relevant services. Location of the device and the user's identity, the user's context and consequently the other details of the user that are stored in the phone like contacts, electronic diary, personal preferences, and prevailing identities; sensitive details like bank information and health related records are being sent to the advertiser or used by the service provider for accurate profiling of the user. All major search engines like

Google, Yahoo etc. and the social networks like Facebook and Twitter etc. have mechanisms to track the location and other personal details of their users. Diana Pouliot – director of mobile advertising at Google – revealed that one-third of all Google searches via the mobile web pertain to some aspect of the searcher's local environment [4]. The service providers are collecting as much information as possible from the mobile device users without their knowledge to ensure revenue. In November 2013, Google agreed to pay a $17 million fine to settle allegations that it secretly tracked web users by placing special digital files on the web browsers of their smart phones [5]. Mobile service providers like AT & T are entering into tie ups for Shop alerts and has put the privacy and hence the security of the user at risk [6] Figure 1.1 shows the privacy threats and their sources.

Most of the smart phones available today, have a built in capability to disable the location information manually. However, the users cannot obtain those location based services in those cases. Also, the feature just allows to turn off the display of location, while the other contacts are still available to the LBSP or the adversary.

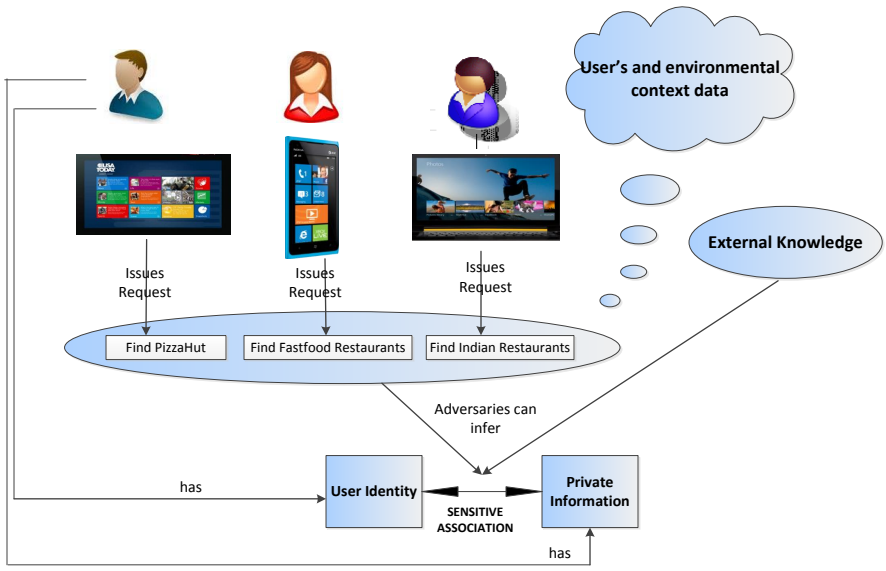


Figure 1.1: Privacy Threats

The threat to breach of privacy has significantly increased due to the increased number of applications that reveal the user's personal data and context to various service providers / advertisers. Due to proliferation of smart mobile devices and advances in geo positioning and wireless communication technologies, many developers are creating novel applications to pro-

vide *Location Based Services (LBS)*. These LBS providers may not be equipped with very large resources. We realistically assume here that there are very large numbers of users who use these services. The mobile device users can obtain a wide variety of location based information, like restaurants at specific locations or events at a location at specific time, and leverage these services to extend the businesses for a competitive advantage for m-commerce and ubiquitous provisioning of their services [7]. The extensive deployment of LBSs not only opens the doors for adversaries; but also endangers the location privacy of the users and exposes the users to abuse, based on the knowledge of the linkage between the two [7]. Also, as shown in [8] , the service providers are collecting other private information of the users (intentionally or unintentionally) which is being sent to the advertisers for the purpose of profiling. Table 1.1 shows the type of information that is collected by the service providers and the percentage applications (LBS providers) from the 1100 Android applications (in 2011) tested by the authors [8] that collect such type of information.

Table 1.1: Number of applications requiring access to resources containing sensitive data and access to Internet [8]

Resource Type	**# of Applications**
phone_state	374 (34.0%)
Location	368 (33.5%)
Contacts	105 (9.5%)
Camera	84 (7.6%)
Account	43 (3.9%)
Logs	38 (3.5%)
Microphone	32 (2.9%)
SMS messages	24 (2.2%)
history and bookmarks	19 (1.7%)
Calendar	9 (0.8%)
subscribed_feeds	2 (0.2%)

As can be seen from Table 1.2 alarmingly large number of service providers are accessing third party Analytics and Advertising (A&A) modules (among the 1100 Android applications and 605 sensitive application) and Table 1.3 shows the information they send to the A&A providers based on the analysis done with 110 Android applications.

Table 1.2: Number of service providers accessing A&A modules [8]

A&A Modules	All 1100	Sensitive 605
admob.android.ads	360 (33%)	225 (37%)
google.ads	242 (22%)	140 (23%)
flurry.androids	110 (10%)	88 (15%)
google.android.apps.analytics	91 (8%)	66 (11%)
Adwhirl	79 (7%)	67 (11%)
mobclix.android.sdk	58 (5%)	46 (8%)
millenialmedia.android	48 (4%)	47 (8%)
qwapi.adclient.android	39 (3%)	37 (6%)

Table 1.3: Number of applications (110) communicating to A&A server any information, IMEI number or Location information [8]

A&A Destination	Any	IMEI	Location
*.admob.com	57	0	11
*.doubleclick.net	36	0	0
Data.flurry.com	27	2	15
*.googlesundication.com	24	0	0
*.mydas.mobi	23	0	0
*.adwhirl.com	21	0	0
*.mobiclix.com	17	10	6
*.googleanalytics.com	17	0	0
tapad.jumptap.com	6	0	0
Droidvertising.appspot.com	5	0	0
*.mojiva.com	4	0	0
*.greystripe.com	2	2	0

These statistics give a glimpse to the issues of privacy protection for the LBS user and is indicative of the cause of concern over privacy protection of the user. The purpose of information collected from the user and sent to the A&A server is to gain additional revenue as has been shown in [9]. **This is our main motivation for proposing this novel integrated framework to achieve *Identity* privacy, *Context* privacy, and *Query* privacy comprehensively without depriving the service provider of the sustainable revenue made from *Context Aware Privacy Protecting Advertising (CAPPA)*.**

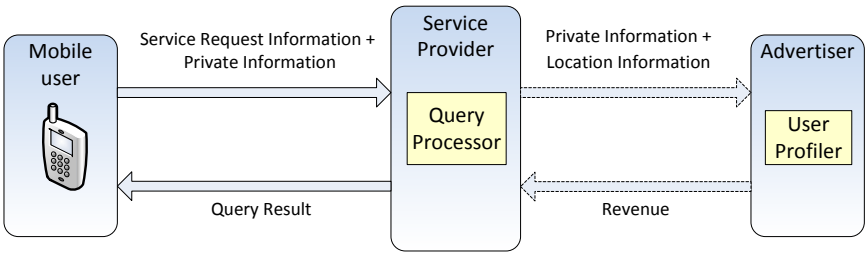


Figure 1.2: Block diagram of basic communication between various agencies

With Google Play Store alone having around 800,000 applications till date and still growing [10], the 10% applications imply that over 80,000 applications are gathering information about the users, beyond just the location information. Figure 1.2 shows the block diagram of basic communication between the involved parties.

The major privacy threat in use of such applications is through the exposure of the spatio temporal characteristics correlation of the query with the user. We believe that the threat is of the highest order when the adversary is able to link all the above mentioned components: the identity, the location, the time of query and the original query. We define these four parameters as privacy protecting parameters. The linkage between these parameters implies simultaneous knowledge of all these parameters. To elaborate this point, consider that a mobile user A wishes to find location for a restaurant (query Q) at location B at time t. The inference that an adversary can make from the knowledge of the user's location B or that the user has initiated a query Q may not be as serious as when the adversary can infer about the user's identity, the location and the time of the query generation. The information that someone has queried about a restaurant may at most reveal the dining preferences of the anonymous person since the identity is not revealed, however, including the time of query as late night may allow the adversary to infer that the user is at home (if typically the user is at home at night.). Also, it might be possible that the user may have issued the query without an intention of going to that place. We now demonstrate that sometimes even in a sensitive case (private information) the knowledge of the user's identity and the location may not be potentially dangerous. For example, the knowledge that user A is querying for the location of a cancer hospital (location B) does not indicate about his/her health situation. The user could just be passing by, or may be visiting someone. So the effort / cost associated with anonymization process may not be necessary. However, if the linkage between the identity, location, query and the time of query

is known there is a larger potential for the breach of privacy and can be potentially dangerous. Another, not so serious but nuisance causing scenario that is possible from this type of linkage is more targeted advertising. For example, the knowledge that the user A has initiated a query regarding medicine for specific diseases, say lung cancer (query Q) from a cancer hospital (location B) is likely to reveal more about the user's health status and may lead to more targeted ads /coupons from the medical suppliers or more seriously misleading information regarding curative medicines. There are three types of adversaries. First type is honest but curious adversary [11]who follows the protocol diligently; but may try to obtain more information than needed by providing forged input data. The second type of adversary is semi honest adversary [7] is basically honest but curious and may attempt to get more information leading to a possible association between identity and location. The third type of adversary is the global observer or malicious adversary who has access to all the information and wishes to use that information for malicious purposes. We wish to expand the horizon of protection as defined in the existing solutions.To address this issue, our focus will be on ensuring that the honest but curious adversary or semi honest adversary will not be able to establish a link between user A, location B and the time of issue of query Q.

Thus, it is imperative to have "privacy protection mechanisms" in place that will protect the identity, location and other context details of the mobile user obtaining the context based services. If any of this information is available in isolation, the threat is not as much as when the linkage between them is revealed. All the solutions offered till date, with the exception of two [9, 12] to the best of the author's knowledge, have attempted to provide mechanisms to protect the privacy but starving the service providers of the revenue that they can generate from the advertising. In the discussion to follow, the query request and the service request will be used interchangeably.

Multiple approaches have been proposed in the literature which will be discussed in detail in the next chapter. A brief overview of the some techniques follows. The approaches that are offered use:

- ♢ Anonymization of identity and / or location,
- ♢ Obfuscation of location using transformations,
- ♢ Use of cryptographic techniques,

- ♢ Modification of the architecture,
- ♢ Access – control based techniques,
- ♢ IP - hiding techniques
- ♢ Modification of the OS to incorporate privacy protection.

Among the current solutions offered, the earliest solutions proposed, ensured the privacy through the use of de-identification or use of pseudonyms, providing anonymity for identity "***k - anonymity***" [13, 14, 15, 16] or location [17, 18, 19]. The basic idea proposed in these solutions is to create k queries from k – users (real or fake) instead of a single query from the user to mask the user as *one of the* k users generating those queries. The other approach is to generate a cloaking region or a minimum bounding region, such that there are k users in the region. This decreases the probability that the service provider can identify the exact user in the bounding area from whom the query has generated. Larger the value of 'k', the higher is the level of anonymity provided [13]. However, in the process of generation of 'k' queries to mask the user's identity, the bandwidth usage increases, causing network congestion thereby reducing the quality of service (QoS) from the service provider. This is especially true in the current scenario where the number of mobile device users and the users of location based services has increased exponentially.

The other commonly suggested technique is obfuscation of the location information[19, 20]. The suggested techniques, most of them using Hilbert space transformations, are computationally intensive and the information sent may not be in the form required by the service provider to service the query. The third technique involves the use of cryptographic protocols [21]. This technique also poses a large overhead on the users' device for computation and requires trusting the service provider, which is a questionable assumption. The IP hiding techniques involve routing the information through various servers [22, 23] which sometimes may be difficult to practically implement. The Access - Control based solutions [24, 25] involves releasing the information based on the role played by the user / provider. This needs extensive maintenance of databases of the roles and the access rules which is a major drawback, given the fact that mobile devices have limited capabilities. Modification in architecture [26, 27] or modification of the operating system [8][9] are other proposed approaches but are practically

difficult to implement since the users may not be prepared to modify the well familiarized operating system (OS).

All of the above solutions can be classified into two types of architectures, one using Trusted Third Party (TTP) servers acting as mediator shown in Figure 1.3.

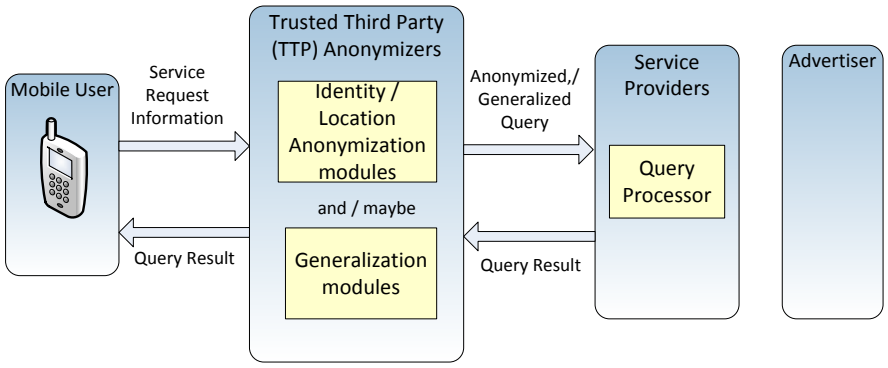


Figure 1.3: Communication model with Privacy using Trusted Third Party (TTP) Server

The second type does not require a TTP server to act as a mediator and the device itself performs the function of providing the privacy protection by transforming the query at the user's device itself as shown in Figure 1.4. The obvious issue with the model in Figure 1.4 is that the device requires a much higher computational capability. It is well known that the computational capability, although on rise, is limited when one is working with mobile devices.

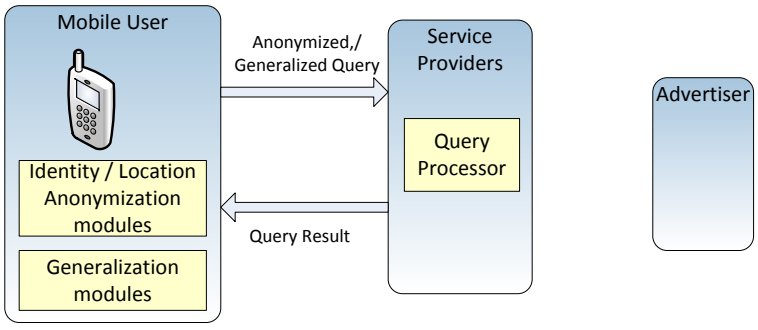


Figure 1.4: Communication model with Privacy without using a Trusted Third party Server

There are two types of attacks that we consider are possible. In communication attacks an ad-

versary gets access to the message streams en route to the servers. The database attacks are when an adversary gets access to the server and hence to all databases stored on that server. For the purpose of proving privacy protection, we consider two most common communication attacks: the snap shot attack and the replay attack. During the snapshot attack, the adversary may get access to any one message string at a time. The adversary based on this particular query string or a sequence of snap shots of query strings from one user tries to infer the link between the user's identity, the location, and the time the query and the query. In case of replay attacks, it can be assumed that the adversary somehow manages to decipher the message string format and injects a large number of fake messages in hope to infer the links. Also, the adversary may inject large number of colluding dummy users into the system.

Table 1.4 briefly summarizes the issues that exist with the generic models listed. The issues with specific models are discussed when those specific models are detailed in the next chapter.

Table 1.4: Issues related to existing generic models

With TTP server	Without TTP server
Typically used in solutions using 'k' anonymity as a technique for privacy protection	Complex calculations, may not be feasible for limited capacity mobile devices
Solutions requiring all users to send location information can cause network congestion	When location resolution is reduced, the service provider may not guarantee accuracy of results
Since traffic is directed to a single server, the message is susceptible to snapshot attack	Since traffic is directed to a single server, the message is susceptible to snapshot attack
Amenable to Replay attack	Amenable to Replay attack
All Information with trusted server, so if server is compromised, all details of user are exposed to the adversary (database attack)	For collaborative solutions, it is difficult to find other users to collaborate at instant the user wants to use a service
None or accurate information is diverted to the advertiser, which might affect revenue of the service provider	None or accurate information is diverted to the advertiser, which might affect revenue of the service provider

The significant challenges associated with quality privacy protection during the use of location based services by a mobile device user are as under:

♢ The major issue in use of location based systems is the implicit trust necessary in the

service provider. Since it has been shown that the service providers are collecting the information not necessarily relevant to processing the query [8] for personal gain and /or revenue, this issue needs to be addressed. The user information, not relevant to the query processing, should not be shared with either the service provider or the advertisers. Most of the existing solutions use a trusted third party mediator. If the TTP Server is compromised, the privacy of the user is also compromised. The computational capability needed in solutions not using the mediators, may not be satisfied by the limited capability of the mobile devices.

♢ Most of the apps available in the market are free or are available at a very minimal cost. This is because the service providers obtain sustainable revenue from sharing the information collected via servicing process (with some additional information not required for processing the service request) with the advertisers [8, 9]. Given the fact that extensive privacy safeguards must be in place for the user, it is also necessary to ensure that the service provider's revenue model is protected **[8]**.

♢ As the number of queries increase with an increase in number of subscribers, the network traffic to the service provider is expected to increase. This is likely to deteriorate the Quality of Service (QoS) (response time will increase) of the service providers. The number of subscribers that the service provider can service will reduce. This is especially true since most of the service providers (application developers) are small businesses with limited bandwidth capacity. If the service provider can service more than multiple users with a single query, it will improve the QoS of the service provider.

All the above mentioned solutions assume that the service provider is willing to accept the different formats in which the systems will provide the information for processing the query. To illustrate this fact, some of the solutions propose to replace the location coordinates by a minimum bounding area. If the solution provides minimum bounding region instead of the location coordinates, the service provider may not be able to process the query. During the query parsing phase, when the query processor is unable to find appropriate fields, the query processor will not process the query correctly.

Some solutions suggested use of a modified operating systems. The users may be unwilling to modify the standard OS to incorporate privacy. As seen in both the existing models, there is no

information, or if there is any, it is inaccurate information that is sent to the advertiser. This is likely to affect the revenue the service provider can obtain from collecting and disbursing the private information to the A & A servers, especially the information that is utilized for accurate profiling of the user, so they can be served relevant advertisements. Also from Table 1.4 we have seen that both the models are not safe from replay attacks. While the model depicted in Figure 1.2 is likely to increase the network traffic (sending k queries instead of one query to provide k *anonymity*) thereby reducing the QoS of the service provider since the service provider may not be able to service all the queries due to his limited resources. We define QoS of the LBS provider as ability to service the customer within the desired response time of the query. (The service providers are the application developers and may not have large number of resources or would not like to waste their resources by generating additional processing responses and contribute to the traffic requiring higher bandwidth). The larger the number of users the LBS provider can serve with fewer number of service requests, the better is his QoS. If the TTP servers send a bounding region, the LBS provider may be able to respond only to range queries. It may also happen that while parsing the query, if the LBS provider encounters the MBR coordinates instead of location coordinates, the query may be dropped. In case of models of type shown in Figure 1.3, most of them use some kind of spatial transformations, performed on the mobile device itself, which may require some collaboration from the location server or may provide location coordinates that are approximate resulting in less accurate response to the query.

We propose a novel CLOPRO framework (**C**ontext C**lo**aking **P**rivacy P**ro**tecting framework) that preserves the privacy of the user while addressing some of the limitations and challenges mentioned above in the existing solutions without compromising the service provider's revenue model. In order to protect the linkage information, i.e. information regarding all four privacy protection parameters simultaneously, from the adversaries and also by the LBS provider (whom we assume to be untrustworthy as shown in [8]) we define a safety cocoon around the mobile device users. We define the safety cocoon to encompass the mobile device users, the CLOPRO framework comprises of two servers that are non-colluding and a secure channel where the communication between them is secured through encryption. Please note that we assume that the LBS providers may also be adversaries (honest but curious type) and one needs to protect the privacy of the user from the service provider too. We may use LBS providers and service providers interchangeably in the following text.

The CLOPRO framework is motivated by the following shortcomings in the existing approaches:

1. If a single centralized server is used for the purpose of anonymization, it may create a serious security and privacy threat if the server is compromised since the server will have all the location as well as query information. It also becomes a single point of failure.

2. None of the existing approaches, to the best of author's knowledge, have viewed privacy protection comprehensively: protecting the identity, the context and the original query simultaneously and avoid exposure of the link between them.

3. If the server has to create a minimum bounding region (MBR) for the purpose of k-anonymization, the servers need to be aware of all the users in that area. Thus, either the servers track the users' location or the users provide their location information continuously. The continuous communication for notification of location may put excessive load and /or network overhead. Also, some of the users who are tracked may not be using a specific context based service. This hampers the privacy of the individuals.

4. Almost none of the available solutions, with exception of couple of them [9, 12] have considered privacy protection from the service providers' point of view. The revenue of the service provider needs to be protected if the solution has to be a economically attractive one.

Our framework makes following novel and unique contributions:

♢ We propose an INTEGRATED framework that preserves the *Identity* privacy, *Context* privacy and *Query* privacy while protecting the revenue model of the service provider.

♢ With the assumption that there is large number of users having similar queries, a clustering technique is employed to cluster similar queries from users in a particular location area at specific time instance window, giving the 'k' anonymity protection [13] to the user so the service provider can service multiple users with a single query. This improves the Quality of Service (QoS) of the service provider since the service provider has to process fewer queries to process large number of users.

♢ The CLOPRO framework achieves query generalization process offline, improving the

query response time. The *Query* privacy protection is achieved by reducing the granularity of the query .

♢ As mentioned earlier, the reason for soliciting more information than required from the user by the service provider is for revenue generation [9]. In our model we effectively address this issue by integrating the relevant advertisements / coupons to the users in the result-set. Our model does this without the need for the service provider getting the details of the users and the link between the location and the query of the user. The only other solution to the best of author's knowledge requires modification of the operating system. The proposed solution protects the revenue model of the service provider without the need to modify the operating system of the mobile device.

♢ In the proposed framework CLOPRO, the technique of caching results improves the service request response time in repeat queries.

CHAPTER 2
RESEARCH TRENDS IN PRIVACY PROTECTION IN LOCATION BASED SERVICES

In this chapter we enumerate and discuss the current efforts that relate to Privacy Protection in Location Based Services (LBS) / Context based services (CBS). We have classified the methods based on the techniques employed, as seen in the Introduction section, and also provide a perspective on the current trends and gaps in the research leading to further work.

2.1 Anonymization based Techniques

Although **anonymity** is defined as "*lack of outstanding, individual, or unusual features; impersonality*" in oxford's dictionary [28], in context of privacy protection, it has been used as ***"the state of being not identifiable within a set of subjects, the anonymity set***" [29]. This category of techniques is used for protecting the identity of the user, location of the user or both. The simplest technique is to use pseudonyms in place of name or other identifying information. However, it is possible to decipher some sensitive information about the user from the combination of other uniquely identifying fields called quasi-identifiers. It is not sufficient to just replace the explicitly identifying information with pseudo values. A variety of other techniques discussed in the following paragraphs have been proposed to overcome this issue.

The seminal work done in the area of '$k - anonymity$' is by Sweeney [13]. The work was to provide a formal protection to the identifying sensitive data that, in a shared data environment, would identify a specific record based on the inferences. The anonymization of identity is achieved either by suppressing and / or reducing the granularity of data. This is done using generalization of the data in quasi-identifying (explicit attributes along with the combination that can uniquely identify individuals), fields in the publicized database. This makes it difficult to identify a specific record from a set of '*k*' records. The authors have set certain policies,

which, if not complied to, can make the records in the database vulnerable to re-identification attacks, which is the possibility of being able to identify the owner using more than one shared databases. This study is important since the methodology developed is generic in nature and is very widely used. Also, the term '*k - anonymity*' coined in this work has become synonymous with a whole gamut of techniques that provides protection by being indistinguishable among k other records (in database) or k users. In medical field or financial information when the data is shared with other agencies, it is important that data does not contain identifiable information that can put the person, whose data it is, to risk. The objective of algorithms developed by the author of that work, is disclosure control, to identify and limit disclosures in released data. The method starts with identification of the quasi identifiers. Once these quasi-identifiers are identified, some part of the identifier is modified using noise (like some string ' * '), that makes more than one tuple indistinguishable from the at least 'k' other tuples in the database. So unless the ' * ' is known, it is impossible to link this information with specific individual.

Record	Race	Birth	Gender	ZIP	Problem
1	Black	1965	M	0214*	Short breath
2	Black	1965	M	0214*	Chest pain
3	Black	1965	F	0213*	Hypertension
4	Black	1965	F	0213*	Hypertension
5	Black	1964	F	0213*	Obesity
6	Black	1964	F	0213*	Chest pain
7	White	1964	M	0213*	Chest pain
8	White	1964	M	0213*	Obesity
9	White	1964	M	0213*	Short Breath
10	White	1967	M	0213*	Chest pain
11	White	1967	M	0213*	Chest pain

Figure 2.1: Example of k – anonymity, where k= 2 and QI = {Race, Birth, Gender, ZIP} [13]

So if k = 2, there exist two tuples in a database that have exactly the same value for each of it's fields making them indistinguishable. Figure 2.1 illustrates one such case clearly for k = 2 and quasi-identifier (QI) set consisting of {Race, Birth, Gender, ZIP} in case of a medical record data. As can be seen from the figure that it is impossible to identify which one of the individuals, amongst details given in record 1 or record 2, have Short breath and which one has Chest pain with just the suppression of the last digit in ZIP field. Thus, every record in the sample

dataset has at least one more record that has exactly the same values except in the last column.

Sweeney further discusses the known attacks against "*k-anonymity*". The Unsorted matching attack (based on order of tuples in released information), Complementary release attack (when one release does ensure *k-anonymity* but subsequent release can reveal in identity) and Temporal attack (Linkage due to matching of tables released at different points in time). A formal treatment used in his paper analyzes the different cases and demonstrates with a proof of how the data is protected. The paper deals with the protection of the identity in stored databases and has not considered the privacy of the query itself or the communication attacks like snap shot or replay attacks.

Machanavajjhala *et al* [30] showed that although there is extensive work done in '$k-anonymity$' area, it still does not guarantee anonymity against the background knowledge attack. It also can leak information from the created groups due to lack of diversity in the sensitive attribute. They have proposed a technique of '$l-diversity$' where anonymity is preserved if the anonymizing block is '$l-diverse$' i.e. if it encompasses at least l "well –represented" values for the sensitive attributes in the table then anonymity will be preserved. To support the process, they have analyzed an ideal notion of privacy termed as Bayes - Optimal Privacy that involves modeling background knowledge as a probability distribution over the attributes. It uses Bayesian inference techniques to reason about privacy. This work does protect the tuples stored in the database; the solution offered does not protect against any communication attacks like snapshot attack and replay attack. The research also does not have any algorithm for *Query* privacy and also to protect the service providers' revenue model. Many other variations include [31] where they have proposed a generalization process of 'm-invariance' that limits the privacy breach in republication of microdata. Masoumezadeh, Joshi [32] have proposed an alternative (k, T) anonymity property that provides anonymity to the users' query against an attacker who is aware of the issuance of the user query within a time window. Many other techniques like defining proximity, diversity, spatial and temporal resolution etc. are also discussed in the literature related to this area of research. Note that all above techniques may work well for protecting the identity of user but have not considered location and/or *Query* privacy protection and so do not provide a comprehensive privacy.

Inspired by initial works of Sweeney *et al* [13], the technique of '$k-anonymity$' is used ex-

tensively in LBSs for location privacy. In these techniques, the user sends his /her location to a third party centralized trusted anonymizer. The anonymizer knows 'k' other users in the vicinity of the user. The server generates a k - anonymized cloaking region that covers not only the user but also $k-1$ other users [14, 21]. Figure 2.2 depicts the concept of $k-anonymity$.

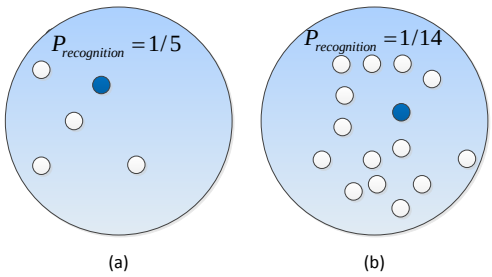


Figure 2.2: Concept of 'k' anonymity

The methods propose that instead of the exact location coordinates, the service provider receives the cloaking region as shown in Figure 2.3, also called as *Minimum Bounding Region* (MBR) of the user. The advantage of the above schemes is that the maximum probability of user being discovered $(P_{recognition})$ is 1/k ($P_{recognition}$= 1/5 in Figure 2.2(a), $P_{recognition}$= 1/14 in Figure 2.2 (b) and $P_{recognition}$= 1/3 in Figure 2.3).

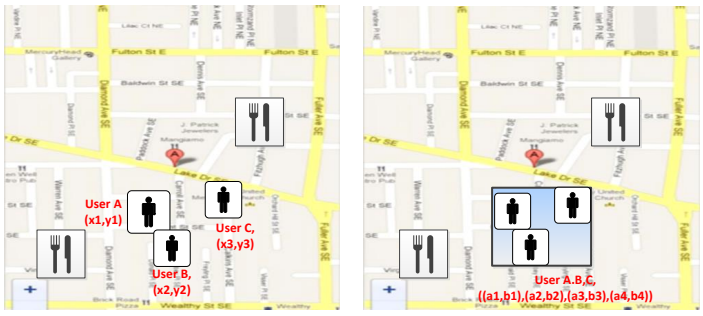


Figure 2.3: Concept of Minimum Bounding Region (MBR)

Larger the value of 'k', the safer the user is, since possibility of discovering the user reduces considerably. Varieties of techniques, some of which are discussed later, have been used to create a cloaking region or the Minimum bounding region (MBR). However, there must be at least 'k' subscribers (real or fake) that are in the vicinity of the user in question. This may be possible in dense areas, but may be a difficult requirement to satisfy, in a less densely popu-

lated area (hence the need for fake users). The cloaking region size may vary depending on the number of users in a particular area. If the area is densely populated, the cloaking region may be very small which can again make the user location traceable. This means that the location cloaking area is population density dependent. Another issue is that other users may not want to relay their location information to maintain privacy of someone else, especially if they are not using LBSs (another reason for using dummy users). One cannot ignore the possibility of the third party anonymizer being compromised. In this case there is a larger risk since the entire database with the personal details of the subscribers will be available to the adversary. Non scalability is one major short coming with this technique. If large numbers of users are requesting LBS and a large number of users (k) are used for creating a cloaking region, the communication traffic can be voluminous, creating bottle neck. There can also be a large processing overhead on the LB server. Since the query is processed via the single anonymizing server, it is susceptible to snap shot attacks and replay attacks.

MBR has been used to create a cloaking region that also can provide *Location* privacy along with *Identity* privacy. Pioneer work in protecting the privacy with respect to the providers of LBSs has been discussed by Gruteser and Grunwald [19] and Mokbel *et al* [33]. In [19], the authors have presented middleware architecture and a set of adaptive algorithms that provide the resolution necessary for E-911 services and yet provide anonymity. Gruteser and Grunwald provide a formal metric for location anonymity, a quad tree based algorithm that reduces the spatial resolution for specified anonymity constraint. In the scenario presented, the authors have differentiated the LBSs on the following three dimensions: frequency of access, time-accuracy/ delay sensitivity, and position accuracy in three typical scenarios of driving condition monitoring, road hazard detection, and a road map. They have considered *Location* privacy as one that can give clues to an adversary to the private information such as medical information, religious affiliations, if an adversary learns about the locations an individual has visited. The key idea is to give a degree of anonymity that can be maintained in any location, irrespective of the population density at that location. The quad tree data structure has been used to maintain the user locations. In the Interval Cloak algorithm by Gruteser and Grunwald for spatial discretization, the area around the individual is subdivided into smaller areas until there are k_{min} subjects in the area. To achieve temporal cloaking, the basic idea is to delay the request until there are k_{min} subjects in the area chosen for the requester. The mobile devices have been simulated using the traffic vehicles. Snapshots have been taken every hour to measure

the resolution. The readings taken have been for determining the area versus the resolution, the relative frequency of occurrence versus the resolution dependency of spatial resolution and mean anonymity constraint. The time of arrival of the user is obtained using the triangulation method. The readings have been taken multiple times and mean values have been noted. Their objective of the research was to demonstrate the implementation of the model using spatial and temporal cloaking and prove that the algorithm is able to provide privacy protection. The paper also demonstrates that the required protection is provided in spite of adversaries. The authors have successfully demonstrated how all the known adversary attacks can be handled when the Interval Cloak algorithm is applied to obtain the Interval Cloak to maintain '$k - anonymity$'. However, the application is successful only for uniform distribution of users but this may not be a reasonable assumption. The solution offered by Gruteser and Grunwald handles only *Location* privacy. Also, the continuous update from all the users will increase the traffic heavily, leading to traffic congestion. The solution will not work well with a small number of users and does not demonstrate how the offered solution will guard against the snap shot and replay attacks.

In Casper [33], the authors have also used the pyramid data structure, where each node has a pointer to its parent, to maintain the locations of the registered users. The registration is with a third party anonymizer server to whom the user provides a privacy profile. This structure is very similar to the quad tree data structure like used by Gruteser and Grunwald. Assume the user lies in a cell 'c' in the lowest layer of pyramid structure when the user initiates a query. If this cell consists of sufficient number of users (k), then the bounding area of 'c' is returned, else the cell size is expanded horizontally and/or vertically depending on which expansion results in satisfying the condition of '$k - anonymity$'. If this still does not satisfy the condition the TTP anonymizing server retrieves the parent of 'c'. This process is done recursively. The success of this implementation depends on how uniformly the users are distributed. Also, if the user leaves the cell, the query must be dropped.

In another approach by Yao *et al* [34], the authors have shown that creating clusters of all users to create a Minimum Bounding Area (MBR) achieves high resilience to location privacy threats. The architecture shown in Figure 2.4 uses trusted third party server

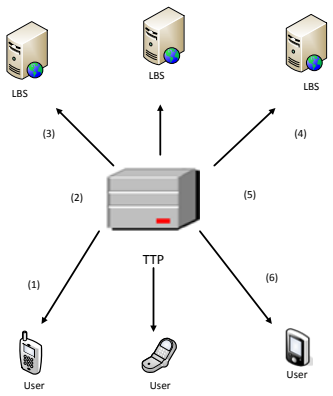


Figure 2.4: Architecture used by Yao *et al* [34]

The original message m_s from the user is defined using parameters of the user id (u_{id}), queryid (n_{id}), location coordinates (x, y), the anonymity level (K) and the content of the query (Ct).

$$m_s \in S\colon m_s = (u_{id}, n_{id}, (x, y), K, Ct) \tag{2.1}$$

The message sent from TTP is of the form:

$$m_t \in T : m_s = \ (u_{id}, n_{id}, X : \Phi(cx, 1/2W_{MBR}), Y : \Phi(cy, 1/2H_{MBR}), Ct) \tag{2.2}$$

The basic idea used is to divide a region into a grid and count the number of users in a grid. If the number of users satisfies the desired 'k' anonymity, the coordinates of grid are notified to the service provider as MBR coordinates. The algorithm proposed by Lin *et al.* works as follows: Initially to create clusters, a simple '*k means*' algorithm using Euclidean distances is used. If the number of users in a grid is larger than 'k', the cluster is divided into a smaller grid. If number of users in a grid is less than the desired 'k', then the neighboring clusters will be combined. The complexity of building the original clusters is shown to be ***O(n lg n)***. They show that clustering can happen in real time and the clusters size and MBR coordinates can change as users leave and join the specific areas. It is unclear from the paper as to whether each of these users in the cluster is expecting a LBS or not. Also, whether the service provider will be able to service the query based on the information sent by the TTP server is not known since the paper assumes that the service provider will service the query in any format that the trusted third party sends it. The major drawback in this approach is that all the users have to report their location continuously, which can be a major network overhead. In addition, the method of choosing the original grid may affect the performance of the system.

Variations: One of the techniques uses dummy users instead of real users to avoid the problem of subscription of large number of real users for anonymizing [35]. In another technique proposed by Lu *et al*, Privacy Area Aware Dummy based Location Privacy (PAD) [36], the idea is to create a virtual circle or a virtual grid defined by the number of locations (k) and minimum coverage area (s); $< k, s >$ privacy requirement as defined by the user. This is necessary to avoid having a very small cloaking region where user density is large (exposing the user) and very large cloaking areas where user density is sparse, thereby increasing the points of interest required to be sent by the service provider. The technique proposes to send a set of locations and a query predicate, that is maintained the same as that in original query. Figure 2.5 (a) shows an example of Privacy Area Aware Dummy generation using Circle based approach where k = 9 and 'pos' is the actual user location. All the positions of the real and the fake users are constrained in a circle of radius 'r', centered at a random position 'r'. All positions are distributed such that all ϑs are equivalent. The upper bound on r is determined using equation (2.3)

$$r = \sqrt{2s}/(k.sin(2\pi/k)) \tag{2.3}$$

and r_{min} is given by equation (2.4).

$$r_{min} = \rho.r \text{ where } 0 < \rho \leq 1 \tag{2.4}$$

If $\rho = \sqrt{3}/2$ the resulting privacy area can be ensured not be smaller than $3.s/4$.

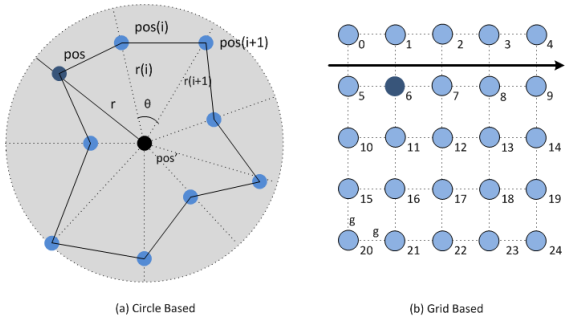


Figure 2.5: Privacy –Area Aware Dummy generation examples [36]

Figure 2.5 (b) shows an example of Privacy Area Aware Dummy generation using Grid based approach with k = 25 and s = 5 and the actual user is at vertex 6 . In the grid based approach,

a uniform square grid with '*k*' vertices and area equal to '*s*' is virtually constructed. The user position is one amongst the '*k*' vertices at the actual position of the user. Although this reduces the need for actual '*k*' users transmitting their location information all the time, the shortcoming is that if the locations are restricted, the identity of the user can be discovered by knowing quasi identifiers like the electronic diary or other information stored in the device connected to a WI FI. Also, the network traffic is increased by a factor of '*k*' since '*k*' queries are going to be generated instead of just one query.

Since determining the *k* can be a difficult proposition, Xu *et al* [18] defined a "safety level" based on the ratio of the geographic region under consideration and the number of nodes inside that region. The larger the ratio, the lesser (and hence) better is the *Location* privacy resolution. The *k* can then be based on the desired safety level of the user. In case of mobile users, the challenge would be to calculate these cloaking boxes without having to reveal the correlation between various boxes. Ghinita *et al* have also used this approach of using the virtual circle or virtual grid covering these locations [37]. One of the techniques suggested the use of fragmented cloaking regions instead of continuous cloaking regions [38]. Chowdhury *et al* [12] have proposed an inclusion of an additional parameter 's–$proximity$' to enhance privacy by ensuring 's' other users having similar static profile when creating the minimum bounding region. The authors have defined anonymity set and also considered static profile parameters while creating the anonymity set. They have given a formal treatment to prove that probability of privacy protection is more when '$s - proximity$' is taken into consideration as against simple k nearest neighbors (kNN) used for privacy protection.

One of the other techniques proposed by Li *et al* [38] verifies the protection to privacy for the trajectory of the user as against just the current location. The technique demonstrated that the set of users used for creating a cloaking region may be moving in different directions thereby creating a possibility to identify the users with sequential information obtained by the adversary. This also provides *k - anonymity trajectory protection* to the concerned user. However, all k-1 users may not be moving in the same direction and cloaking region will get larger, thus not remaining as useful for location specific queries. The authors propose the use of fake queries when the user in the cloaked region has moved out of the subsequent cloaking regions. In another proposed solution [39] the authors suggest the use of footprints of user (user's location sample collected at some point of time) instead of other k-1 users, to avoid multiple users, re-

laying their location information to the server. This will also help in reducing the bandwidth wastage. Also, if there are a large number of footprints from different types of users in a particular region of interest, it may be difficult to correlate the identity of a specific user. In [40], Xu *et al* have suggested exploiting the knowledge of location history of the user to create sequential cloaking regions.

Figure 2.6 shows an example of how foot prints are used to determine cloaking regions to achieve '$k - anonymity$' in continuous trajectory model. The darkest foot print is of the actual user in motion, the additive foot prints shown in gray and white, both from different trajectories, are added to ensure that there are always 'k' users footprints (here k = 3) in the consecutive cloaking regions.

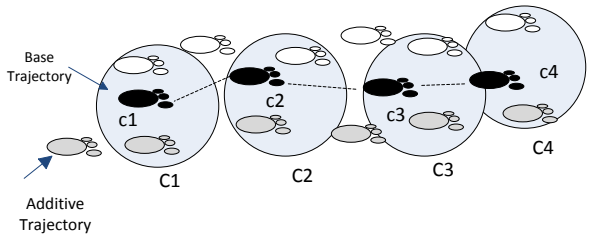


Figure 2.6: An example of k- anonymity in trajectory model [40]

However, in this technique, one needs to create a data base of footprints at all locations and have an additional burden of processing these data bases.

If the linkage between the user query and location as well as the user identity has to be concealed, it is necessary to take into account the *Identity* privacy, *Location* privacy as well as the *Query* privacy. In a preliminary paper by Mano *et al* [41] the authors have considered location as well as "private attribute" privacy. They have used a third party server (matcher) that is used for cloaking as well as matching the user's interests with the results obtained from service provider. The user raises a query and specifies the '*k*' desired level of anonymity along with two additional parameters '*l*' (length of side of square cloaking region) and '*s*' (a parameter specifying the distance within which the results are applicable). The matcher constructs all possible '*k*' anonymity cloaking regions. Based on the value of '*l*' for each of cloaking regions, if none of the cloaking regions have value of '*l*' less than the value of '*l*' specified by the user,

a message regarding cloaking as failed is issued to the user. The authors have also defined certain similarity measures for the area as well as the different types of attribute parameters. The anonymization operation is performed on the location coordinates as well as the private attributes of the user. The authors have chosen advertisement service where the advertisements have been specified with respect to the attributes of the profile chosen. Although, this model is a good proof of concept and emphasizes the need to provide privacy to the private attributes, there are many shortcomings in the model. The attributes chosen will change based on the service that the user avails. Their paper illustrated that entire user information regarding the location, and private attributes is stored in a single table in plain text. So, if the adversary hacks into the trusted server, a single database will allow the adversary to get entire information of the user. The model is not secure under replay attacks. The k - *anonymity* can also be achieved using peer – to – peer communication [21, 42].

Lin *et al* [43] have proposed an Enhanced Clustering Cloak algorithm (ECC) and have used a third party trusted server to provide *Location* privacy as well as *Query* privacy. When a user asks for a particular LBS, the information is sent to the service provider as a message query that includes information about the user, the location, and some other details required to process the query. It may also contain additional information not required to process the query. *Location* privacy implies preventing the disclosure of the location information collected by the adversary or the attacker to identify the user of the service (i.e. link the user and the location by the service provider). The *Query* privacy, in their paper, focuses on preventing the attacker from figuring out the identity of the users through analyzing contents of the query. The authors have achieved this using the k means clustering for *Location* privacy and hierarchical clustering for *Query* privacy (*Query* generalization). In this study the authors have attempted to develop a combined clustering algorithm (ECC) that will allow the user of a LBS to achieve *Location* privacy as well as *Query* privacy that satisfy the 'k- *anonymity*' and '*BK - Diversity*' (l - *diversity* [30] and m - *invariance* properties [31] are satisfied). The experimental set is evaluated for cloaking times and attack avoidance. The authors have used Simulation based comparative study in which the performance (in terms of cloaking times and attack avoidance) of several well-known algorithms (Local – k, Neighbor k, Casper, Hilbert Cloak and Interval Cloak technique for clustering) is compared to that of the Enhanced Clustering Cloak (ECC) algorithm. This being a simulation study, a random sample of moving cars was simulated and requests for the service was randomly generated from each. The accessible population was

the available system (Simulation of Urban Mobility) to simulate the moving mobile users (the authors used moving cars analogous to the moving mobile device user) and a query generation engine was developed to generate random queries using convenience random sampling. When the number of queries increases the traffic increases. So the authors initially, developed the algorithms for *k means* clustering and hierarchical clustering. The *k - means* clustering was performed to cluster at least *k* users that have a query from a clustered area whose coordinates are sent to the service provider. The cluster size was increased or decreased to ensure that a minimum of *k* users were in the bounding area. The hierarchical clustering was used find a minimal query subset that covers all those queries generated from those *k* users. To test these algorithms, a map was downloaded for facilitating the movement of the cars. Also a query generation engine that randomly generated queries was developed. Well known algorithms were coded in Java and were applied in case of the randomly generated movement of cars as well the randomly generated queries. Based on the number of clusters created on application of each of the known algorithms was measured. The simple recording of measurement was done. Average values of the cloaking times were recorded. However, standard deviation has not been calculated to understand the nature of variability. The cloaking times and number of clusters created have been shown, but no insight is provided into the total service time for each of the *k* - user's to get the service even if it provides better privacy, it will be difficult to apply it in real life since the user's location in mobile environment is always changing. Also the bandwidth usage has not been measured in this study.

The outcome indicated that the algorithm does provide higher privacy level than most of the well-known algorithms. The algorithm worked best in terms of attack avoidance and but had comparable or slightly better performance for cloaking time. The complexity of algorithm is not calculated. The possibility that the trusted third party can be compromised is not taken into consideration in this paper. Also, since a single server is performing both these activities once the user sends the query to the server, the system is amenable to snap shot attacks and replay attacks.

2.2 Cryptography Protocol Based Techniques

The second set of techniques use some type of cryptographic protocols to achieve the privacy. The specific transformations are known only to the clients, so the server will process the user query without the ability to decipher the exact location of the user. The server in this case cannot offer guarantee of result for obvious reasons like inaccurate results [44, 45]. Some solutions may use encryption techniques like SSL at the anonymizer or at the private device (Mobile phone) and then send the details to the service provider. This helps in encrypting the user information and the location information to provide privacy / security or other cryptographic techniques like Kerberos to provide authentication, authorization and channel encryption. Another technique that is used is the use of Private Information Retrieval (PIR) protocol [21]. Using the PIR approach, the user can access the data base without revealing which item they are retrieving. When the PIR technique is used in data bases, the entire data base is sent to the user which is then queried in the privacy of the individual device. This approach does not require the use of a trusted third party server. The basic operation of the PIR Framework is shown in Figure 2.7.

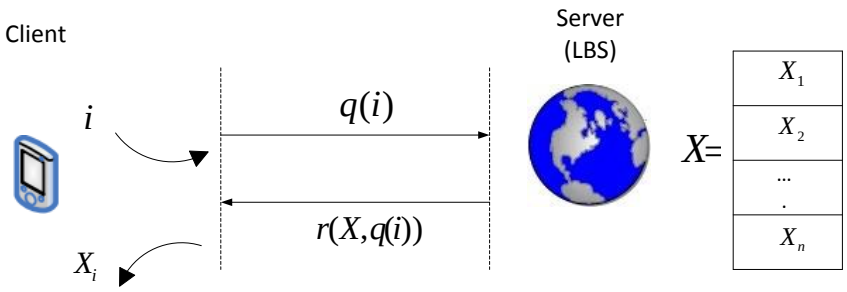


Figure 2.7: PIR Framework [21]

The database in Figure 2.7 is an n-bit binary string X. Assume that the user wants to find the value of X_i. The user sends an encrypted request *q(i)* to the server. The server responds with a value *r(X, q(i))*. The client then computes X_i from the response of the server. It has been proven by the authors that it is computationally intractable for an attacker to find the value of i, given *q(i)*. The privacy is preserved since the request has been sent in an encrypted form.

There are two techniques demonstrated by the authors Ginita *et al* in this paper, one involving storing the location indexes as a kd - tree that retrieve approximate NN query results as shown in Figure 2.8 (a) and overlapping a grid over Voronoi diagrams that retrieve more exact NN query results as shown in Figure 2.8 (b) respectively. Example shown in Figure 2.8 (a) depicts the results received by '*u*', the querying user, where the LBS contains four points of interest p_1, p_2, p_3, p_4. The LBS generates a kd-tree index of the POIs and partitions the space into three regions A, B, C in an off line phase. During the phase of query processing, the user finds himself in space A, utilizes PIR to request all points of interest. Upon the receipt of the POIs in region A in encrypted form the user decrypts the obtained information and obtains $\{p_4\}$ as his NN. The server does not know that the user is in region A. The method can be used with a variety of indices. The authors have also implemented the algorithm using Hilbert space and R-Tree data structures. The results may not be accurate but are quite close.

To obtain the exact results for NN query the server computes the Voronoi diagram for each of the points of interests (POIs) in the pre-processing phase. Each POI p_i is assigned to its Voronoi cell; by definition, p_i is the NN of any point. A regular grid of arbitrary granularity is superimposed on the top of the Voronoi diagram by the server. Information about the Voronoi cells intersecting the grid is stored in a data store. In the example shown in Figure 2.8 (b), D1 stores p_4, whereas C3 stores $\{p_3, p_4\}$. When the user queries the LBS server, the client retrieves the grid granularity to calculate the grid cell that contains the POI (i.e., C2). Using PIR the user requests the contents of C2. The user receives p_3, p_4 in encrypted form. Then p_3 is calculated as exact NN.

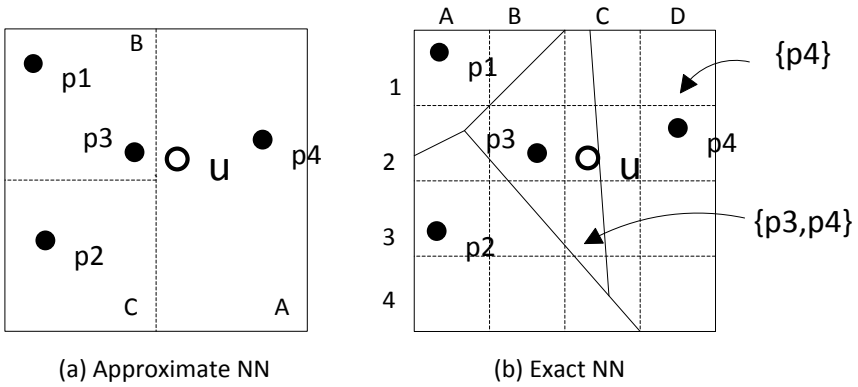


Figure 2.8: PIR Example [21]

The process requires complex computation and hence incurs high computation cost (if one decides to find a way to send limited information in case of a very large database) and the communication overhead may be impractical or infeasible in a mobile environment. These types of protocols are good for the *Location* privacy, but do very little for the *Identity* or *Query* protection.

2.3 Obfuscation / Transformation based Techniques

Now we discuss the set of techniques proposed in the past that use certain transformations on the query to obfuscate the location / user information. Obfuscation may be through real, fixed, chosen locations from some special ones like road intersections [46, 47] or dummy locations generated randomly [35, 48].

One of the commonly used techniques [20] utilizes space filling curves as one way transformations to encode the location of users and the point of interest into an encrypted space. This query is then evaluated in an encrypted space. Since the distance properties of the original space are maintained, efficient evaluation of queries in transformed space is possible. On receiving the results, the users can inverse the transformation efficiently using the stored information which is available with them and thus the users are protected from malicious servers. In [14], Kalnis, *et al* suggested Hilbert Cloak that uses the Hilbert space filling curve

for mapping the two dimensional space into one dimensional values. The data structure used is annotated B+ tree. The one dimensional sorted list is then partitioned into groups of 'k' users through the Hilbert Cloak algorithm. To find a particular user, the cloaking algorithm finds the group to which the user belongs and returns the minimum bounding rectangle of the group as the cloaking region. Note that all the users in a particular group will have the same location coordinates. Thus, the Hilbert Cloak ensures privacy of users for any distribution of users. This technique is useful for static snapshot *Location* privacy. Similar techniques of obfuscation have also been enumerated by Wishart *et al* [49]. Many modifications to these basic techniques have been enumerated in literature; a couple of them are discussed below.

In PRIVE´ [50], Ghinita *et al* have proposed a novel algorithm hilbASR that uses a decentralized architecture to preserve the anonymity of the users issuing spatial queries to LBSs, PRIVE´. The bottleneck caused by the centralized techniques, both in terms of anonymization and location updates are avoided in PRIVE´. The basic idea of hilbASR defines a total order of user locations using Hilbert space filling curve as shown in Figure 2.9.

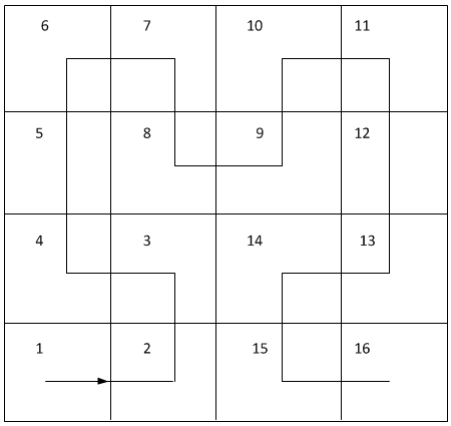


Figure 2.9: Hilbert Space [17]

The transformation using the hilbASR algorithm is shown in Figure 2.10 (for 'k' = 3 and 'k' =4). The algorithm proposes a data structure to store users information in the order defined through the Hilbert space filling curve. The data structure is partitioned into blocks of k users. The obvious problem with this solution is that the 'k' has to be the same for all users. Note that the last block may have (2.k -1) users. The algorithm returns the MBR computed considering the position of the users that are in the same block as the issuer.

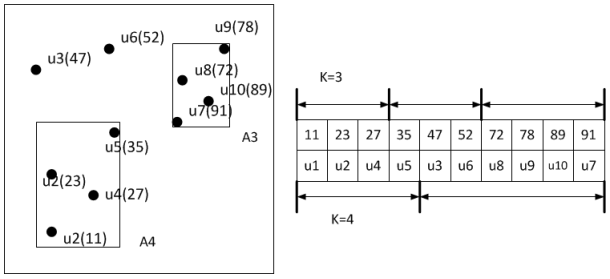


Figure 2.10: HilbASR Transformation [17]

Yiu *et al* [17] have evaluated a technique that will send relatively accurate points of interest. When multiple users are querying for point of interest (POI), the service provider uses kNN query to return the k POI's nearest to the user's current location. To deliver *Location* privacy, one of the methods is that the user's location is cloaked using spatial cloaking where the exact location is replaced by a larger area making its impossible to reconstruct the exact location from the knowledge of cloaked region. The other alternative method uses some kind of transformation of location coordinates such as Hilbert ordering. This may result in a not-so-accurate query result, in case of spatial cloaking, since larger than required number of POI's will be returned. This increases the processing and communication costs if the cloaking region is large and the cloaking region has a large number of POI's. For transformation, the query may need to be evaluated in a transformed space, since the service provider cannot decode Hilbert's values, this ordering will need to be uploaded in the server before the query can be responded to.

To address this issue, the authors Yiu *et al* designed a technique called SpaceTwist in which the query fetches the nearest POI's incrementally until the desired accuracy is obtained. The model eliminates the need for a trusted third party server. The server indexes the dataset of POI's using a R-Tree data structure and supports incremental nearest neighbor retrieval. The client retrieves initial response from a location close to, but not the original location of the user and then uses INN (Incremental Nearest Neighbor) query processing iteratively. Incrementally, POI's are fetched until accurate query result can be reported. The authors have reported that the client executes an algorithm to fetch k nearest objects from the server using a fake location. To reduce the traffic, the server packs multiple results in a packet and sends it to the client. This also helps in increasing the *Location* privacy. In the SpaceTwist model, the server knows the exact query although the results set contains multiple POI's not relevant to query.

This means that although the *Location* privacy is provided, the user is not guaranteed a *Query* privacy. Also, the number of queries to be processed for one actual query will be dependent on the number of POI's, distance of the actual location from the initial location, desired k. This can increase the communication cost and also the processing cost.

"CAP: A Context Aware Privacy Preservation System for Location Based Services" by Pingley *et al* [51] uses spatial transformation. In the study, they propose an idea in which the 2 D (lat –long) location coordinates are projected to a 1D space such that every point in 1D space has homogenous context and adjacent locations also remain close after the transformation. Once this transformation is complete, a random perturbation using known noise distribution is introduced. This perturbed value is then mapped back again to the 2D coordinates which are then transmitted as the user's location to the LBS. The performance measures (outcome variables) are the accuracy of the query answer, the N-Confidentiality (No malicious server can distinguish between any two locations in a region of population N) and communication quality of the service. Random selection of 800 points of interest and about 1000 different co-ordinate points lying in areas of varying Road densities (such as the downtown, the market or the rural areas) were used as possible locations of the user. The experiments were simulated and the values for the retrieval of VHC maps, the 2D perturbation distances, extra traveling distances were noted. The accuracy of location perturbing component for a scenario on issue of the top 10 query for the nearest point of interest was also noted. If the perturbation is small i.e. the noise is small, then the coordinates of point of interest will be near the actual location. But if the random perturbation moves the location coordinates far from the desired point, the query results may not be relevant. Also, if the service provider has to provide the results back, when the location is not an accurate location and so the results may not reach the desired user. These techniques however are difficult to be extended to other contexts since they are specifically dealing with the locations only. Also, in the enthusiasm to enhance the privacy, obfuscation could lead to generalization to the extent of rendering the services unusable.

2.4 IP hiding techniques

When the mobile devices are used for receiving context aware services, there is a communication vulnerability where the traffic from the user to the service provider and the response can disclose the IP of the user. Cryptographic techniques are used for privacy protection of the traffic to and from the service provider. The location information can be deciphered from GPS coordinates and also the IP of the device sending the requests. So, if along with the techniques for reducing location resolution, when the IP is also encoded, it may be possible to dissociate the location of the user from the service request. The two systems proposed are, Tarzan [22] and Onion routing [23]. In Tarzan, the concept of the proxy server is used where the request goes out through the proxy, ensuring exact IP of the user remains hidden. However, this system requires additional server mimic which substitutes the user's IP with the pseudonym corresponding to its domain. Although not mentioned explicitly, many of the solutions using third party servers effectively use this mechanism. In Onion routing, there is a set of proxies through which the query travels are re-encrypted at every proxy server. Since the proxy route is unpredictable, the system provides good IP hiding. However, this solution may fail if one of the nodes is compromised. In one other technique detailed in [52] there is an additional layer (lighthouse) created that dissociates the user and the service provider through a portal. The portal does not know the requesting user and the user does not know which portal will be used for providing this service. The privacy protection will be at the cost of service response time assuming that nodes in the server hierarchy will never be compromised.

2.5 Access Control based Techniques

The privacy of the user can be assured if one controls the quantum and the type of information that is revealed to the service provider. This category of techniques is used to control the amount of access to private information that could pose as a threat to privacy. As discussed in [53] one can classify them into two classes: Discretionary Access Control (DAC) and Role-Based Access Control (RBAC). The first category requires that the users define generic policies about the data/information that can be accessed by each service provider based on certain conditions.

Some work has recently been done by encoding policies as digital certificates using SPKI/SDSI certificates using RSA based signature scheme [54]. Houdini framework developed at Bell Labs [55] has provided an extensive formal framework to represent, transform, integrate and share the dynamic, heterogeneous data based on the defined policies. In these models, the information will be disclosed based on certain policies laid down by the user. Atluri *et al* in their paper [24] suggested creating a unified index structure SSTP tree that holds the mobile user's dynamic information (past, present and future) along with the authorization as specified by the user and the profiles of moving objects. The second category is more general in the sense that the policies are dependent on the roles the user or the service provider is in, rather than the generic policies. This means that the user may want to share some information while playing a particular role like in a college or at work or on vacation and the policies are defined for the role the user is playing at the moment. Rigid RBAC may not be suitable in this situation, but a modified version incorporating spatio-temporal components have been included in some work [25]. The problems with these techniques are that a separate structure has to be maintained, and given the limitations of memory and the processing power of the devices, it may affect the processing speed of the devices. These policies are also based on the trust which exists between the users and the service providers. Since this is based on the expectations, there is always a possibility of misuse. Also, these techniques, by themselves, are not sufficient for complete privacy protection.

2.6 Architecture based solutions

Attempts have been made to look at architectures that will inherently be privacy protecting. One of the earliest attempts was made by Drost [26] who modified and extended the WASP (Web Architecture for Services Platforms) architecture to ensure privacy. He incorporated a privacy layer between the service manager and the semantic service description acting as a broker. All the communication from the device to the service provider will move through this layer to ensure that the privacy policies are applied in the communication. UniCOSM middleware [27] incorporates privacy protection with the use of mechanisms for secure access to the context data and ubiquitous services. It also distinguishes between the physical (spatial characteristics) and the logical dimensions (current activity, capabilities of devices) of the user

and the device. Braun *et al* [56] have proposed a multi-agent service ware framework that will assist the service provider in creating personalized services in a secure and a privacy aiding method improving the acceptability of the users. In his master's dissertation and in the paper published further, Jagtap [57] has presented a framework using Semantic Web Technologies to specify the users' sharing preferences through high level declarative policies. This has been verified on the University of Maryland campus network as reported in his paper. Aggregation of the heterogeneous data, privacy protection of the users information as well as the inferences that can be drawn from the said information, enable better control over information flow over the network. Riboni *et al* [53] have proposed a complete framework for privacy protection. They ensure that most of the vulnerabilities from user perspective are taken care of. However, in their method, the privacy issues and their repercussions from the service provider's perspective have not been given consideration. The privacy requirement for the push type of services is not taken into consideration. Hong [58] in his work has proposed an infrastructure 'Context Fabric' to ensure users' privacy in the Context Aware Systems. The system provides support for ensuring the privacy in context aware applications by letting users define the privacy tags for each context tuple which may slow down the entire process.

In our earlier work [15] we proposed a framework which uses a single Trusted Third Party server that would act as a Context Proxy Server (CPS) to provide the privacy to user as shown in Figure 2.11. This framework has a single trusted server and so if the CPS is compromised, the user details will compromised. Also, the revenue model of the service provider is not protected in that model.

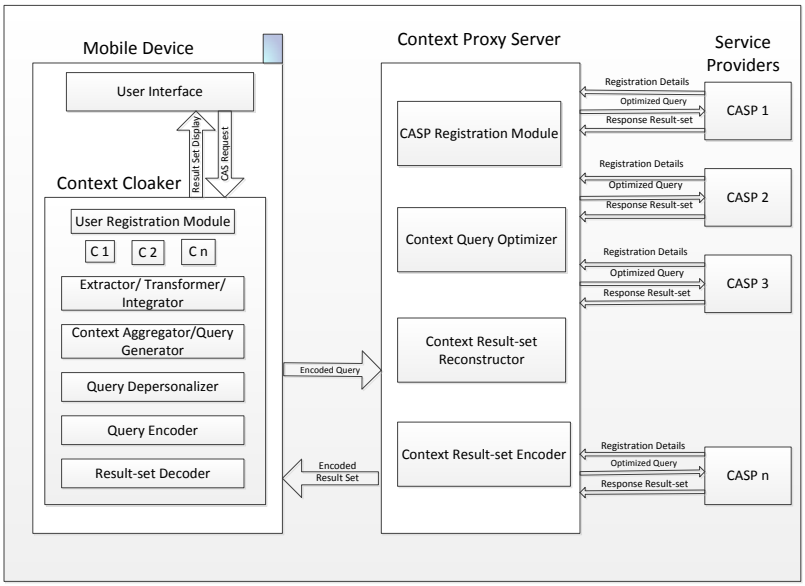


Figure 2.11: Context Proxy Server based framework [15]

In the paper by Guha, *et al* [11] the authors have addressed another major issue. While using LBSs from a mobile phone, there is a constant threat that the information other than the intended by application is available to the adversaries who can use this information for malicious purposes. This model 'Koi' is specifically useful when the servers are in the cloud. The goal of 'Koi' is to provide the location functionality to the applications that need it, while ensuring that no third party, such as entity other than the user and the service provider (application developer), is in a position to learn the association between a user's identity and their locations or other attributes. It is assumed that if a user is using a particular application, the user trusts the service provider to not use the information for malicious or advertising and marketing purposes. Thus, it is necessary to create a platform for the application developers to be able to provide the service while protecting privacy of the user information. The architecture of 'Koi' is shown in the Figure 2.12.

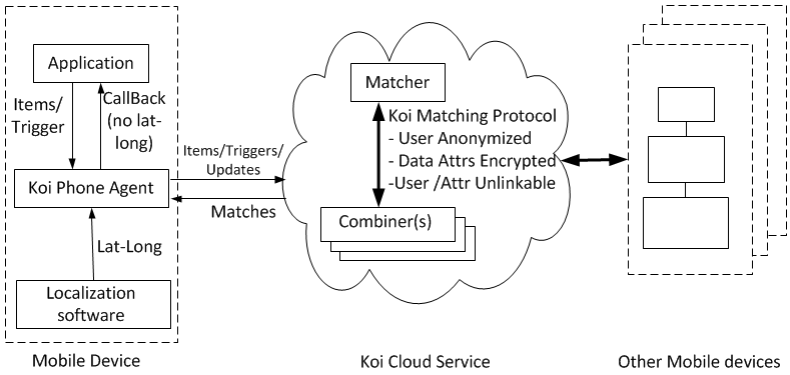


Figure 2.12: Koi architecture [11]

The method used in the technique proposed by Guha *et al* is to separate the information between the matcher and the combiner with different components encrypted and so with information from either one of these, it is not possible to establish a link between the user and the registered attribute by the user. Single Subject Experimental methodology is used since the authors have taken some individual applications and verified that the link between the user and the attribute (could be location) or a link between the user who has registered more than one attributes is not identified by the attacker. They also verified that neither the combiner nor the matcher can find these links unless they are colluding. The authors have chosen an application developed by them, which provide navigational service and verified that honest but curious combiner or matcher cannot establish the links. A sufficiently large amount of trace of mobility data was used to confirm that the privacy was preserved. The initial step was to develop the platforms and the protocols required to support the 'Koi' platform. The protocols used by the authors were:

- ♢ **Registration** of the user and the attributes of the user,
- ♢ **Matching Attributes** based on the event of a trigger where the combiner then finds the match between the associated attributes for a given registration.
- ♢ **Combining Matches** to check in the 'Matching' set if the registration id with the matcher and an encrypted registration id based on trigger are found.

Formal 'Koi' model was developed using pi –calculus and was manually verified by them and also verified using ProVerif on the developed application to identify the linkages between the

user and the attributes by a combiner or a matcher using Macro and Micro benchmarks. The authors successfully demonstrated that it is not possible for either the matcher or the combiner to violate the linkages between the user and the attribute or the multiple attributes of a specific user unless they are colluding. The basic flaw in the above approach is in the assumption that if the user is using a particular application then the application is trusted. This is contrary to the case since there are multiple references that show that the applications collect information about the users without their knowledge which is used by the third parties for either malicious purposes or for marketing and advertising purposes [8, 9]. The other questionable assumption made in this work is that the developers will be interested in the use of this platform. If the business model is based on collecting the information and the linkages, the possibility that the developers will adopt the platform seems flawed. This study has not noted the time required for acquiring the service and the bandwidth usage even if the privacy is provided if the combiner and matcher servers are not colluding. From this point of view, there should have been more than one dependent variable. The model has not taken into consideration the *Query* privacy. This model seems to be secure against snap shot attacks and replay attacks since the information messages are encrypted.

2.7 Other techniques

There have been attempts to use some combination of the above mentioned techniques or managing the cache for privacy protection [59]. Chen *et al*. [59] have proposed the use of cache management along with the existing anonymizer to enhance the privacy of the Context Aware Systems. Nearest Neighbor query technique or its variations have also been used by various researchers.

One component that has been missing in all the above approaches is the consideration of the revenue generation for the service provider. The main reasons behind the service provider sending private information to the advertisers is for the revenue the service provider receives from the advertiser, so the ads /coupons can be served to the customers. The first reported work is by Mano *et al* in [41]. Although they have not explicitly mentioned the revenue aspects, they have modeled their privacy protection solution assuming that advertisements have

to be served. Recent work based on this has been reported in a paper by Leontiadis *et al* [9]. The authors explicitly mention the need to provide information to the service provider for the purpose of revenue generation while protecting the privacy of the user.

These authors focus on the fact that accurate customer profiling is a must in the advertising driven market. Some solutions like “MockDroid” or “Apex” which provide privacy are by [60, 61], however, they fail in considering the implications for the market. If rigorous privacy protection mechanisms are in place, it might lead to a collapse of the ad - driven mobile phone market. Free mobile apps request more potentially dangerous information than the paid ones as classified by Android OS organization. When the frequency of warnings increases, the users tend to ignore them.

These authors [9] discuss the market model where the advertisers pay the developer of the app (service provider) with the finances based on the number of impressions of advertisements. The developer includes the code that collects private information from the user, profile and pass on the information to the advertiser. Since the funds increase significantly when the user clicks on the advertisements, there is a strong incentive to generate more clicks per ad impression.

As the revenue will increase when the ads match the interests of the users, the advertisers need to collect as much information about the user as possible to accurately profile the user. This also means that even if a list of permissions is provided to the user, the user may not be aware of how that information is used and also whether the application or the desired service requires those permissions. The current implementation and envisaged model is shown in Figure 2.13

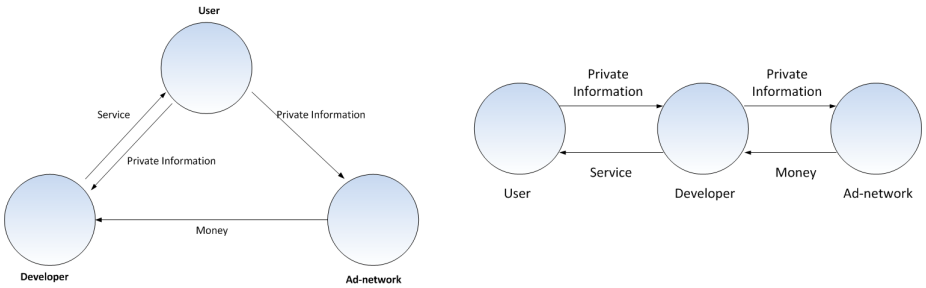


Figure 2.13: Current Implementation and envisaged model to provide advertisers with information [9]

Most of the solutions offered are to block the information and modify it with falsified information. However, repeated use of this technique will starve the advertiser of accurate information necessary for profiling and revenue for the service provider /developer. This could potentially diminish the revenue stream for the developer and will hamper the sustainability of the **free** application market. In response to this situation, the authors [9] suggest decoupling the flow of the profiling information from the privacy control component. The decoupling permits specification of separate privacy policies for the two components of the model: the application providing the actual service and the profiling information. The level of restrictions for the flow of information may be different for each component, may be stricter for the profiling component. This also reinstates the trust of the user while using the application.

The authors [9] propose redesigning of the market to enhance the awareness and incentivize the developers. The authors recognize the main challenge is to establish a causal link between the flow of information to the advertisers and flow of revenue. The proposed solution allows the user to aggregate the number of generated "clicks" to get an estimate of the revenue the developer can get from the use of application and through a feedback mechanism tune the settings for privacy policies for profiling component. These authors also propose the developers to specify the amount of revenue expected to modify the ratio of information flow to profiling component. The major drawback in this model is that all the involved parties, the developer, the user and the advertiser have to change the way they think and operate. Also, many additional mechanisms such as modification in Operating systems and modification to the message processing will need to be in place for the concerned parties to make a switch. Application of peer pressure on the developers to avoid the use of such mechanism is difficult to implement.

2.8 Summary

It has been seen from the discussion above that the issue of privacy protection in LBSs and context aware services has many aspects and they have been addressed in multiple ways. Most of the research has addressed the issue of protecting the privacy of identity (the user and the query), or the location. Recently researchers have realized that it is important to look at both of these together and linking the same to the revenue.

For *Identity* privacy, the approaches used include substituting the identifying parameters with

pseudo values and *'k - anonymity*'. There have been multiple approaches used to achieve *'k – anonymity*'. Some of these were to create a minimum bounding region to include *'k*' other users (real or fake) to fuzzify the querying user. The methods suggested sending *'k - queries*' instead of one, report the region coordinates of an area instead of exact location with *'k*' users (real or dummy) in it, use footstep information to create minimum bounding region. All of the solutions, when looked at in isolation, do not completely preserve the privacy since the location or the quasi identifying information can reveal the identity. Also, when a single server is used to create a minimum bounding region for providing *'k - anonymity*', the server is susceptible to database attack and also to communication attacks like snapshot attacks and replay attacks.

The concept of *'k - anonymity*' has been applied for providing the *Location* privacy too. Methods used for *Location* privacy include perturbing the location coordinates, sending minimum bounding area coordinates instead of location coordinates, applying some type of transformation like Hilbert space transformations or some other transformation. Most of the techniques are computationally intensive, and also the inaccuracy of the location coordinates does not guarantee accurate results in return. When the location coordinates are fuzzified, it is also difficult for the service provider to send the results back to the user.

Some of the solutions require computation by the service provider, especially in the case of the Hilbert ordering which involves additional computation from the service provider's end. If the service provider is not performing the computation, a trusted third party server will be necessary to perform this step in privacy protection. This type of solution, as well as some of the cryptographic solutions are likely to secure the database from attacks.

Some proposed methods have suggested a collaborative effort in providing *'k - anonymity*'. The users have to register with a central server that informs the specific user about other users in their vicinity. These users communicate amongst themselves (may be via blue tooth) and send a query from any random user or create a bounding region and send those coordinates. This avoids the use of a single trusted third party server to create bounding region, assuming that the users may be willing to communicate when they are not users of a LBS. This assumption may not be valid in all the cases making the proposed solutions infeasible.

To protect the ip of the mobile user, some solutions have used concepts of routing and also

multiple routing through proxy servers. This does provide identity protection, but the time involved in the process makes it an infeasible solution option since time is critical in obtaining the location / context based services.

Modifying the records in the database by replacing one or more quasi-identifying fields with some character like a '*' will ensure that there are at least k other records that are similar and will provide privacy to the records in the database. Ensuring *'l-diversity*' and *'m-diversity*', to bring in more diversity in the records of the database will protect the user from the database attacks. However, the problem of communication attacks has not been looked at in these solutions.

Query attack is another aspect that could breach the privacy of the user. The most common solution offered for creating *Query* privacy is generalizing the query (reducing granularity) to obtain a larger set of 'POI' (Points of Interest) from the service provider. This increases the network traffic since many POIs are sent in response to a query of a single user.

Some solutions have suggested the modification in the architecture. Some have suggested modification in the operating system to ensure that no information, that the user does not want to share, is communicated to the service provider by modifying or falsifying the information. This is the best possible solution, however, not many users will be willing to risk modifying their operating system just for these add on services.

Amongst all the solutions offered, most have focused only on the privacy aspect. Many of the solutions assume that the service provider to be a trusted entity which in the past has been shown to be untrue. However, the main reasons why service providers extract and collect more information than needed for the service is for revenue generation. Quite a few of these services are provided free to the user since the service provider is able to generate revenue from the advertisers associated with these services or similar such applications. The stricter the method of privacy protection, the service provider is likely to be starved of the revenue generated from providing information to the advertisers who use it for customer profiling. Except a couple of proposed methods, modification of OS or modification by the service providers in the way they develop their applications, none of the solutions yet offered have considered the aspect of protection of revenue model of the service provider. The stricter the privacy controls get, more

starved will be the service providers for accurate information to generate revenue. None of the existing solutions have viewed at privacy protection comprehensively with minimal or no modifications by the user or the service provider without compromising the revenue expected.

It is necessary to view the privacy protection comprehensively. In the CLOPRO framework, we attempt to create a comprehensive framework that is privacy protecting, does not require the user to use modified OS or requires the service provider to modify the way he develops the application while protecting the revenue model of the service provider. By using clustering for *Location* privacy and abstraction for *Query* privacy, we have ensured that the framework is not susceptible to communication attacks. Also, the "who/what and where" components of the query are processed separately using a distributed architecture. This reduces the vulnerability to database attacks. The framework successfully demonstrates the protection of privacy from the honest but curious adversaries. Table 2.1 gives a bird's eye view of the various technologies from certain representative literature. The explanation of the need for using specific features for comparison is shown in Table 2.2.

Table 2.1: Comparison of various existing Solutions

Models		Performance parameters										
Technique	**References / models**	**Uses TTP Server**	**Location Privacy?**	**Query Privacy?**	**SP is trusted?**	**Query Format of SP maintained?**	**Number of queries processed by SP**	**Caching of Query Response**	**Can protect against Snap shot Attack?**	**Can protect against Replay attack?**	**Can protect against DB attack?**	**Single Point of Failure?**
k anonymity based	Machanavajjhala [29]	√	√	X	Not Considered	X	k querries/ customer (if fake queries are generated) or 1 query/ customer (if location address is changed)	X	X	X	√	√
	Masoumzadeh m-invariance [31]	√	X	√	X	X	k queries / user	X	X	X	√	√
	Gruteser, Interval Cloak [18]	√	√	X	X	Only Range query	1 query / user	X	√	√	X	√
	Mokbel, New Casper [32]	√	√	X	X	Only Range query	1 query / user	X	√	√	X	√
	Pandit, CCPPS[14]	√	√	√	X	Only Range Query	1 query / user	√	√	√	X	√
	Lu, PAD [35]	√	√	X	X	Only Range Query	k queries / user	X	√	√	X	√
	Mano [40]	√	√	To some extent	X	Only Range queriy	one query/ user	X	X	X	X	√
	Lin, Combined clustering [33]	√	√	√	Not considered	X	one query/ k users	X	X	X	√	√
Cryptography based	Ghinita,PIR [20]	X	√	√	X	√	1 query /user	X	√	X	√	X
Transformation based	Kalnis Hilbert Cloak [48]	X	√	X	√	√	1 query / user	X	√	√	√	X
	Ghinita, Privé [50]	X (Uses Peers)	√	√	X	X	1 query / user	X	√	√	√	X
	Pingle, CAP [51]	X	√	√	X	√	1 query / user	X	√	√	X	X
	Yiu, Space twist [16]	X	√	X	X	X	one query/ customer	X	√	√	NA	X
Architecture based	Guha, Koi [11]	√	√	X	√	X	one query / customer	X	X	X	X	X
Modification of OS	Leonitidis[9]	X (Spilts information)	X	√	√	√	split query/ customer	X	X	X	X	√
Comprehensive	CLOPRO	√	√	√	X	√	one query / k customers	√	√	√	√	X
√ ==> Supported	X ==> Not supported											

Table 2.2: Explanation of features in Table 2.1

Features	
Trusted Third Party Server?	Use of Trusted Third Party server to protect the location and query privacy
Location Privacy?	Location privacy preserved? (Link between user and his location at specific time)
Query Privacy?	Exact queryof the user is not known to the service provider. Decreasing the granularity of the query will improve the privacy for the user.
SP is trusted?	Many times SP, for the purpose of revenue, shares the user's sensitive information (link between location and user's identity and the queries they asked at specific times) with advertisers, third parties not involved in processing of the requests.
Query Format of SP maintained?	The Service Provider has a protocol requirement for processing a query. If it is not in same format, the query may not be processed by the Service Provider.
Revenue model of SP protected	Motivation for a Service provider who is providing free services is through revenue model of pushing ads/coupons to the users based on their context.
Number of Queries processed by SP	When large number of queries are to be processed by the SP, the response time increases and the Quality of Service will deteriorate as number of users increase.
Caching of Query Response	When the query is repeated, the TTP can respond without communicating with the Service provider. This will reduce the response time for a query
Protection against Snap shot attack	If the message is intercepted by adversary, is it possible for the adversary to correlate the information and find link between user and his location and the query asked by him at the specific time.
Protection against Fake Queries	If the adversary creates fake queries, is it possible for the adversary to correlate the information and find link between user and his location and the query asked by him at the specific time.
Protection against DB attack	In case the adversary gets access to the Trusted Third party server, is it possible for the adversary to correlate the information and find link between user and his location and the query asked by him at the specific time.
Single Point of Failure	If there is a single server used as a TTP server, failure of this server will stop the process

CHAPTER 3
CONTEXT CLOAKING PRIVACY PROTECTING FRAMEWORK (CLOPRO): AN OVERVIEW

We have proposed a new framework "***CLOPRO – A Context Cloaking Privacy Protecting System***" that ensures the *Location* as well as the *Query* privacy for the mobile user utilizing the services of a Location Based System (LBS). Our system uses Trusted Third Party Anonymizing Servers. To ensure privacy we employ a distributed architecture with two non-colluding servers: one to achieve the *Location* privacy and the second for *Query* privacy. Clustering technique is used for providing *Location* privacy and agglomeration is used for reducing the granularity of the original query. A unique user id and a query id for each query allows for de-identification of the user and the query. We heuristically show that it is not possible for an honest but curios attacker to link the user's identity, location and the query that has been issued at a specific time. Our model also ensures that the LBS provider can be assured of his revenue through disbursement of ***context aware privacy protecting advertising*** *(CAPPA)*.

3.1 CLOPRO Description

We describe the functioning of the proposed system using figures /tables and logical sequence diagrams. The logical system architecture proposed in this work is shown in Figure 3.1. Note that mobile device users are registered users and always communicate with the LBSP (acronym defined later) through the CLOPRO framework (marked in the figure in an enclosure).

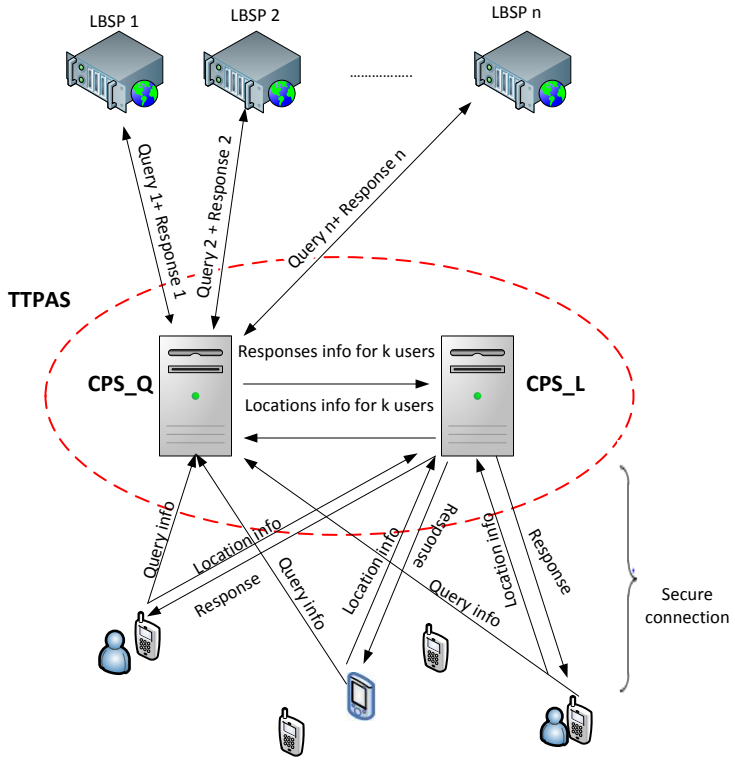


Figure 3.1: Logical System Architecture

For the sake of brevity and for the convenience of referencing, we first summarize the symbols and the acronyms used throughout text for description of the system in Table 3.1. Some additional symbols may be introduced later.

Table 3.1: Symbols and Acronyms used with Descriptions

Symbol/ Acronym	Description
LBSP	Location Based Services Provider
CPS_L	Location Anonymization Server
CPS_Q	Query Anonymization Server
ID	Unique user id
$< ID_m, ID_q >$	Unique Request Reference Pair
k	Desired level of Anonymity
t_s	Start time of the query
t_f	Desired response time of query
δt	Incremental time frame within which queries will be clustered
q_{gen_i}	Generic query for user i
q_{orig_i}	Original query for user i
q_{code}	Code for generic query
sr	Service Request
$sr^{'}$	Transformed Service Request
SR	A Set of Service Requests
$SR^{'}$	A Set of Transformed Service Requests
$\mathbf{X_i}$	Location coordinates of the user
ID_{clust}	Identity of the cluster
ad_i	Advertisement associated with query i handled by LBSP
$coupon_i$	Coupon associated with query i handled by LBSP
$sale_i$	Sale notification associated with query i handled by LBSP
QoS	Quality of Service

The CLOPRO has following five components:

1. Mobile User
2. Location Anonymization Server (CPS_L)
3. Query Anonymization Server (CPS_Q)
4. Location Based Services Provider (LBSP)
5. Advertiser

The Query Anonymization Server (CPS_Q) and Location Anonymization Server (CPS_L) are the two ***non-colluding*** servers that comprise of our CLOPRO framework. Note that the CLOPRO framework is a secure gateway to the untrusted LBSPs. Our earlier attempt at creating a framework for privacy protection comprised of a single trusted third party server. We found that the compromise of that server would result in complete breach of privacy making it inevitable to encrypt all the stored information [15]. So, the processes of who/what and where were split to provide further protection and also reduce the number of parameters that required encryption to provide more privacy protection than that offered by our earlier model.

The top level view of the functioning of the five components, their interactions and the operation of each of the servers is depicted in Figure 3.2.

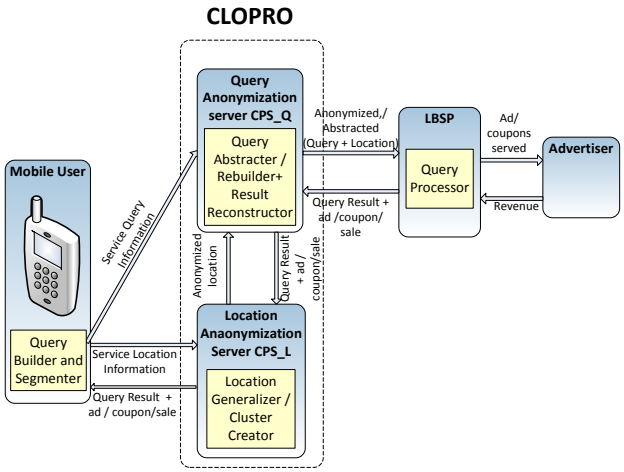


Figure 3.2: CLOPRO framework

3.1.1 Mobile User

The user initially has to download the application. The application contains a module that communicates with the user interface. The user communicates with the system for:

♢ **Registration:** The user registers with the CPS_Q where the user specifies the LBS applications he/she wishes to use, the ad / coupons / sale notifications he is interested in. Demographic information of the user is stored in an encrypted form in User_Information db on CPS_Q server. The Query server in turn generates a Unique identification for the user (ID) sends it to the user along with a subset of the query code table (QCode db) consisting of query codes (q_{code}) for different levels of abstractions of the query (q_{gen}) in the service requests (q_{orig}) for the LBS applications the user has expresed desire to use.

Service Request Processing

It comprises of two steps i) Query Generation and ii) Query segmentation

Query Generation: During the service request process, the user inputs the query (q_{orig_i}), the desired level of anonymity 'k' and the query expiration time (t_f). An example service request from the user is of the form :<Restaurant_Finder, Pizza Hut, 4, 10> i.e. the user wants to use Restaurant Finder application and wants to find Pizza Hut with an anonymity level of

4. The desired query response time is 10 seconds. The Query builder module generates a unique request reference pair $< ID_m, ID_q >$for each service request to uniquely identify the request (ID_q) and the generator of the request (ID_m). However, this component is encrypted for enhanced privacy. The generic query code (q_{code_i}) for the original request as specified by the user is appended to the service request. This $q_{code_i} \in q_{code}$set is received by the Mobile device from CPS_Q after the registration process. Each service request $sr_i \in SR$ where SR is a set of service requests is of the form:

$$sr_i :=< (ID_m, ID_q), q_{orig_i}, q_{code_i}, k, \mathbf{X}_i, t_s, t_f > \tag{3.1}$$

After the modifications by the Query builder module on the mobile device the service request appears as:<(ABC0123, ABC1123), PizzaHut, R22, 4, 38.29880, -85.99302, 2013-05-1412:21:49.780, 10 >. As can be seen from the message string, a unique service request reference pair has been generated (ABC0123, ABC1123) (the reference pair is shown unencrypted here for explanation purpose), q_{code_i} (R22) is appended with the location coordinates ($\mathbf{X}_i$) and query initiation time to the original request of the user.

The module of the application on the users' device generates a unique request reference pair $< ID_m, ID_q >$ to uniquely identify the generator of the query as well as the query of the user. Please note that a new request reference pair has to be generated for each service request of the user. The advantage of this technique is that even if an adversary is able to retrieve snap shot of the query string, linking of the originators of the different requests will not be obvious in case of continuous queries. The module appends the generic query code (q_{code_i}) for the query that the user has generated and splits the message into two components: one that is delivered to the Query Anonymizing Server (CPS_Q) and the other to the Location Anonymizing Server (CPS_L). The new query strings generated are of the form:

The service query information to the CPS_Q:

$$q_{CPS_Q} :=< (ID_m, ID_q), q_{orig_i}, q_{code}, t_f > \tag{3.2}$$

The service location information to the CPS_L:

$$q_{CPS_L} :=< ID_q, k, \mathbf{X}_i, t_s, t_f > \tag{3.3}$$

The processing of each of these strings will be explained in the subsequent sections.

3.1.2 Location Anonymization Server (CPS_L)

The major activities performed by the CPS_L are:

♢ **Location Transformation:** The first distinctive feature of our model for *Location* privacy is that we cluster the requests of '*k*' users who have issued a specific generic query (same q_{code_i}) , issued in a specific time window (between t_s and $t_s + \delta t$ where $\delta t \ll (t_f - t_s)$) and are co-located in the area of specific dimension and the cluster centroid is reported as location for all those k users. As can be seen from the query string sent to the CPS_L, the CPS_L does not receive the details of the actual query of the user, just a code associated with the generic query. So **the CPS_L server is not aware of the user and the actual query of the user that it is servicing**. A subject is considered as location '*k - anonymous*' if and only if the location information sent from a mobile client to an LBSP is indistinguishable from location information of at least *k* -1 other mobile clients [19]. There exists a close relation between the *Location* privacy and the location '*k-anonymity*'. The task of CPS_L is to transform the location coordinates of the various mobile device users co-located in a specific diameter, and have a similar generic query generated in a particular time window making them indistinguishable amongst '*k* -1' other users. Also, this anonymity has to be maintained, regardless of the population density and the number of querying users with same generic query and in a limited time frame within a specific area.

We propose the use of a simple clustering technique to create clusters of '*k*' users each, where the technique guarantees a minimum of '*k*' anonymity. In case a particular cluster does not have the requisite number of users, i.e. the number of users in a cluster is less than the k_{max} amongst the users in the cluster, the algorithm used for transformation by CPS_L generates the necessary number of random users within the cluster to satisfy '*k*' anonymity criteria. For example, if a cluster has 5 users and the maximum requested '*k*' among the users in the cluster is 7, then two random users will be created within the cluster (taking into account the distribution in the cloaked region) and the new centroid, as the location coordinate of the 5 users in the cluster, will be sent to the CPS_Q for further processing of the request. Note here that although the dummy users are generated in our scheme, the network is not burdened with additional requests since only the cen-

troid will be passed to the CPS_Q. This additional step guarantees that the '*k*' desired by the user has been satisfied. It has been shown that higher the value of '*k*', stronger is the anonymity provided to the user [13, 19]. Also, the '*k*' is '*personalized*', in the sense that each user defines his / her '*k*'. Our algorithm creates a cluster of actual / dummy '*k*' users who have initiated a query and hence will not burden the network by sending dummy or fake queries to the LBSP to guarantee '*k*' anonymity. In fact, it is easy to visualize that the technique reduces the traffic to the LBSP by sending a single query that covers queries from '*k*' users, thereby improving the QoS of the LBSP. The CPS_L stores the original coordinates along with transformed coordinates for retrieval while result dispatching.The CPS_L sends these location coordinates ($\mathbf{X}_{clust}$) along with the *set of* ID_q*'s* of the queries that are clustered together.

$$q_{CPS_L \rightarrow CPS_Q,} := < \{ID_q\}_k, q_{code_i}, \mathbf{X}_{clust}, \{t_f\}_k > \tag{3.4}$$

For example, the message string from CPS_L to CPS_Q server will be of the form: < {ABC1123, ASD1234, ADK1762, AXW1222}, R22, 38.0627, -85.37403, {10, 12, 12, 11}> . In this example three requests from users ABC, ASD, ADK, AXW are clustered together (centroid coordinates are 38.0627, -85.37403). The anonymity requirements of these users are (4, 3, 3, 4). Since the maximum k of these requests is 4, the cluster consists of all actual users. However, if there were fewer than 4 user's in the time window with same q_{code_i} then dummy user's within a specific distance would be created randomly to satisfy this k requirement. The cluster centroid would be calculated using location of actual users and dummy users. {10, 12, 12, 11} are the desired query response times of the users.

♢ **Result Dispatching:** Once the result set, along with the ads / coupons / sale notifications have been received from the CPS_Q, the Result Dispatcher module (discussed in Chapter 4) obtains the stored original location coordinates $\mathbf{X}_i$ of the user for the specific query id ID_q and dispatches the results to the appropriate mobile user at correct location.

$$q_{CPS_L \rightarrow user} := < R, \{ad_i\}, \{coupon_i\}, \{sale_i\} > \tag{3.5}$$

The resident module on the mobile device displays these results to the user.

3.1.3 Query Anonymization Server (CPS_Q)

The CPS_Q performs two types of operations : online (Registration and Service Request Processing) and offline (Query Code Generation):

♢ **Registration of user:** When the user registers with the CPS_Q, a unique ID is generated. The user information about the applications (s)he desires to use, the interests etc are stored for further processing in the CPS_Q. The demographic information is encrypted and stored in a User_Information database on the CPS_Q server. It then extracts the subset of q_{code} that the user will require based on the applications to be used and sends it to the User along with the unique identification of the user.

♢ **Registration of service provider:** The LBSP also has to register with the CPS_Q. The process is similar to registration of user, so it generates a unique id (ID_{LBSP}) for each service provider, separates the information it has received from the LBSP regarding the services it provides and their description, the service request formats for each type of service request it can handle as well as the ads /coupons / sale notifications that the LBSP wishes to disburse to the users.

♢ **Generic Query Generation:** The other distinctive task, which ensures the *Query* privacy, is the generation of a set of generic queries for a particular application and assignment of code to each of the generic query. The CPS_Q performs this task offline, there by reducing the time required for the task of query aggregation. For each application $\mathcal{A}$, there exists a set of generic queries (q_{gen}) defined by $\mathcal{Q}$, that covers the entire range of queries that are possible for a particular application. For example, the specific query of "Where is the nearest MacDonald's?" or "Where is the nearest Pizza Hut?" are covered under a generic query of "Where are the nearest Fast food restaurants?". Formally this can be expressed as,

$$\forall \mathcal{A} \exists \mathcal{Q} \ni \mathcal{Q} := \bigcup \{q_{gen_i}\} \tag{3.6}$$

Each original query q_{orig_i} is comprised of a set of features that can be represented as:

$$q_{orig_i} :=< F_1, F_2, ..., F_n >_i \tag{3.7}$$

where F_n is the feature at leaf level (lowest granularity). The q_{gen_i} is then created using some combination of the features $< F_1, F_2, ..., F_{n-1} >_i$ depending on the desired granularity for the query using query generalization technique. Although F_n (leaf feature) is at lowest granularity, all of these features need not be hierarchical in nature.

Please note that each of the q_{gen_i} may cover more than one q_{orig_i} and it is possible that one q_{orig_i} may appear in more than one q_{gen_i} . This is due to the fact that generic queries are generated with different granularity. This set of query codes is passed on to the user when he registers with the CPS_Q depending on the applications he chooses to use. For example, suppose the original queries are pertaining to location of restaurant and the features $< F_1, F_2, ..., F_n >_i$ are: type of restaurant (F_1), cuisine (F_2), name of restaurant (F_3- the leaf feature). Generic queries may be based on the type of the restaurant, the cuisine, the cuisine and the type etc. Let us assume that possible types of restaurants are {Fast food, Sit-in, Sit-in with Bar}, possible values for Cuisine are {Indian, Italian, Chinese, Mexican}, and possible restaurant names (Points Of Interests or POIs) are {Olive Garden, China One, MacDonald's, Pizza Hut, Qdoba, Taco Bell, El Nopal, Tumbleweed, Bombay Grill, Little India, Sitar}. So the generic query of "Where is nearest Sit-in Italian restaurant?" will return in response the location of Olive Garden, whereas the query "Where is the nearest Indian Restaurant?" may return the location of one or more Indian restaurants {Bombay Grill, Little India, Sitar} depending on current location as specified by the centroid of the cluster. Formally, with n_i features, the abstraction process can be specified as ($\oplus$indicates concatenation):

$$q_{gen_i} := \oplus_{i=1}^{n_i-1} a_i F_i \text{ where } \begin{cases} a_i = 0 & F_i \text{ is } absent \\ a_i = 1 & F_i \text{ is } present \end{cases} \text{ in } q_{gen_i} \tag{3.8}$$

♢ **Query Rebuilding:** The next task is to rebuild the query depending on the query format registered by the LBSP. The **Query Rebuilder** module first matches the ID_q to determine the user id (ID_m) associated with each of the ID_q. It then generates a unique id for the cluster (ID_{clust}). If the service provider requires exact location coordinates, the CPS_Q uses the centroid and creates a k Nearest Neighbor query. However, if the LBSP accepts the Minimum Bounding Region (MBR) coordinates, then the CPS_Q creates an MBR using the centroid coordinates and generates Range queries. The CPS_Q does not know the exact location of any of these '*k*' users. The query string that is sent to the LBS

provider may be of the format:

$$sr^{'} : q_{LBSP} :=< ID_{clust}, \mathbf{X}_{clust}, q_{gen_i} > \tag{3.9}$$

or

$$sr^{'} : q_{LBSP} :=< ID_{clust}, \mu(x_1, x_2), \mu(y_1, y_2), q_{gen_i} > \tag{3.10}$$

where $\mu(x_1, x_2), \mu(y_1, y_2)$ will define the coordinates for MBR as per the service request format requirement of the LBS provider. Typically some constant value will be added to create a block of reasonable size. Note that $q_{LBSP} = sr'$ as mentioned earlier. We call *sr'* the transformed format of service request *sr*. The service request sent to the LBSP will look like: $< QWE, 38.0627, -85.37403, "Cuisine =' Mexican'" >$or $< QWE, 38.06, 38.07 - 85.37, -85.38, "Cuisine =' Mexican'" >$

♢ **Result Reconstruction:** Once the service request results are obtained from the LBSP, the CPS_Q then splits the generic result into specific results using the "de-abstraction process" for each query, matches the interests of the specific users (that have been specified by the users at the time of registration) with the query and sends this to CPS_L. The format of this message string is of the form:

$$q_{CPS_Q \rightarrow CPS_L} :=< ID_q, R, \{ad_i\}, \{coupon_i\}, \{sale_i\} > \tag{3.11}$$

The results of the query are stored in a database. The only parameters that the CPS_Q saves is the original query, the query code and the "result-set" returned by the LBSP along with a time stamp. This will improve the query response time when similar query is repeated in the future. The CPS_Q can return the results directly without approaching the LBSP. Since the LBSP updates the CPS_Q with the ads /coupons / sale notifications, associated with products, locations or other points of interests, the CPS_Q can deliver those to the users and send information regarding number of recipients of specific ads / coupons / sale notifications to the LBSP. This is win – win situation for both, the user and the LBSP, since response time to user's query is reduced and LBSP can still get revenue from delivery of ads / coupons / sale notifications. The CPS_Q sends back the number of coupons / ads /sale notifications served to the users without revealing the identity or

location or the query of the user to the LBSP.

3.1.4 Location Based Services Provider (LBSP)

The LBSP also has to download an app to be able to communicate with the CPS_Q server. The Communications between LBSP and CPS_Q are for the purposes of:

♢ **Registration**: The LBSP registers with the CPS_Q by supplying the information regarding the types of services, the description of the services and the service request format required for processing each of these service requests. The LBSP also sends the advertisements (interchangeably referred to as ads), coupons, sale notifications that they may be served to the user along with the location associated with it and the validity period. The CPS_Q sends the unique ID_{LBSP} to LBSP which will be used in further communication. The LBSP updates the information regarding ads / coupons / sale notifications as and when newer details are to be served. The CPS_Q purges the ads / coupons / sale notification once they have expired.

♢ **Result processing:** The final distinctive feature of our framework is *Context Aware Privacy Protecting Advertising* (CAPPA). The LBSP sends the "result set" for the query to CPS_Q as well as the ads/ coupons/ sale notification associated with the query and also associated with the location coordinates that it has received. Note that the LBSP does not know the user, the query or the exact location of the user, yet can successfully deliver the ads / coupons / sale notification to the user based on his / her context. This ensures that the service provider can earn the revenue that he / she could have received on providing the user information to the A&A servers. The response sent back by the LBSP is of the format:

$$q_{LBSP \rightarrow CPS_Q} := < ID_{clust}, \{R, \{ad_i\}, \{coupon_i\}, \{sale_i\}\} > \tag{3.12}$$

There will be more than one points of interests (POIs) returned since the service request comprised of generic query rather than accurate query.

The CPS_Q after matching the user's interests, the query and the ads / coupons / sale notifications obtained from LBSP, sends back information to LBSP regarding the number of ads /

coupons / sale notifications served to anonymized users.

3.1.5 Advertisers

Our CLOPRO framework does not do any kind of communication with the Advertiser. The LBSP communicates with the advertiser about the number of ads / coupons / sale notifications served to users. Based on the prior agreement, the advertisers send the revenue to the LBSP.

The detailed stepwise functioning of each of the servers and the algorithms are explained in the following chapter.

CHAPTER 4

CONTEXT CLOAKING PRIVACY PROTECTING FRAMEWORK (CLOPRO) - DESIGN DETAILS

In this chapter, we discuss, in details, of the functioning of our model and develop algorithms that will provide *Location* and *Query* privacy. Before we define the system formally, we will describe the two basic operational steps in details and then describe the formulation of algorithms necessary for these steps.

4.1 LBSP Registration

The first task we describe is the LBSP registration depicted in the Figure 4.1. The registration process is handled by the CPS_Q server. During the initialization phase, the service provider registers with the CPS_Q server. The LBSP provides with the following information to the CPS_Q server:

$$< LBSP_{name}, ServiceName, ServiceDescription, QueryFormat, ad, loc_{ad}, validity_{ad}, coupon, loc_{coup}, validity_{coup}, saleinfo, loc_{sale} validity_{sale} > \quad (4.1)$$

There can be multiple services provided by the service provider and each of the services will be identified by the name, description and the query format necessary for processing the query. The service provider also specifies the ads, coupons and sale notifications that the provider wishes to disburse along with the location of POI and the time validity for the same.

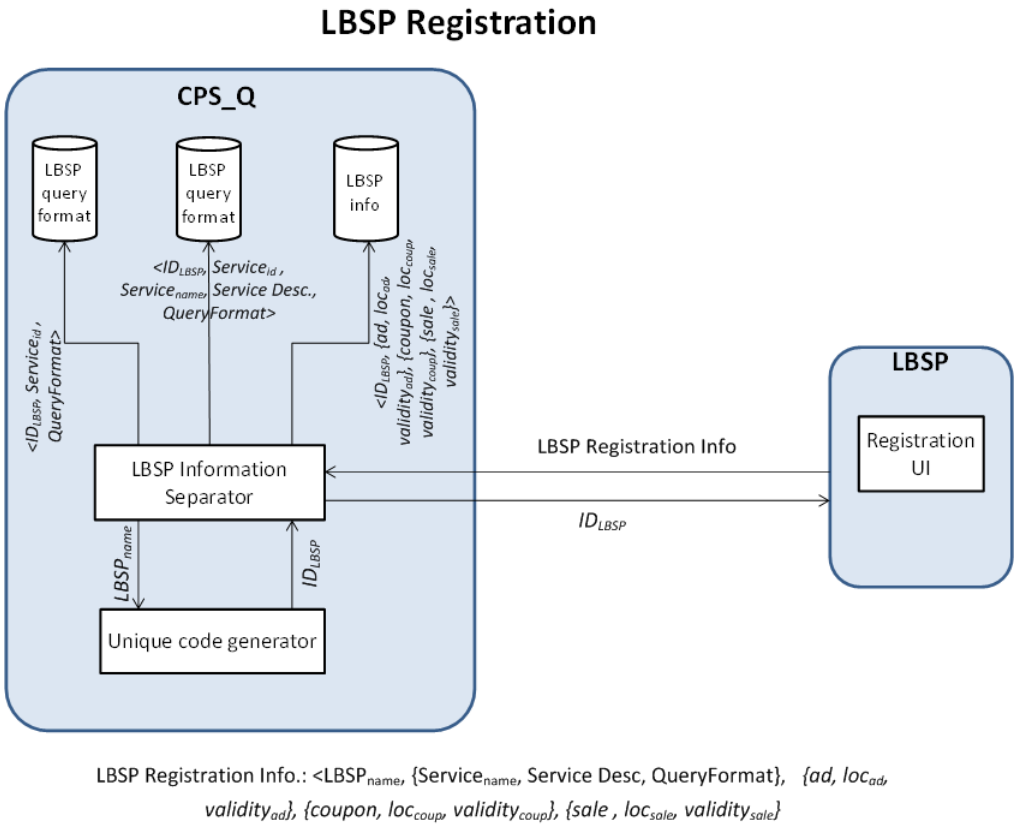


Figure 4.1: LBSP registration process

At the CPS_Q server, a parser at the LBSP_Information_Separator parses this information and stores it in different databases viz: LBSP_Queryformat db, LBSP_Servicename db and LBSP_ Info db, associating with each record a service_id ($Service_{id}$). The **LBSP Information separator** module sends the LBSPname to the **Unique code generator** module. This module generates a unique id, ID_{LBSP} that is sent to the service provider.

The updating of the LBSPInfo db and LBSP_Services db is event triggered i.e. the service provider will update this information as and when there is a change in the available information. The sequence diagram showing the details of LBSP registration process is depicted in Figure 4.2. The top of the sequence diagram indicates the entities involved in the process and also the databases involved. The movement of the messages between the various entities are shown and are organized temporally from top to bottom. So, the first task is the movement of LBSP registration information from the LBSP to CPS_Q. This information is separated by the LBSP Information Separator module that sends the $LBSP_{name}$ to the Unique _code_generator which then returns the ID_{LBSP}. The LBSP Information Separator module then store the Query format in the LBSP_Query format db, the description of the various services provided

by the LBSP in the LBSP_services db and the other demographic information in the LBSP_info db. The information stored in the LBSP_Query format db is used by the CPS_Q server while rebuilding the query. Information stored in LBSP_services db is used by the CPS_Q server while reconstructing the result set to find if there are any ads / coupons / sale notices that the user has expressed interest in obtaining.

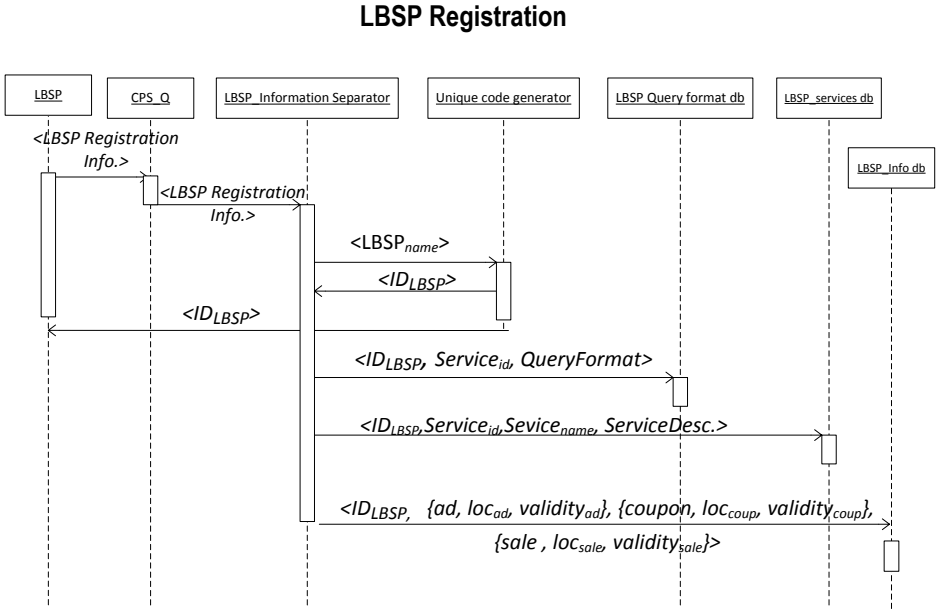


Figure 4.2: LBSP Registration Sequence Diagram

4.2 User Registration

Similar to the LBSP, the user also registers with the CPS_Q server as is shown in Figure 4.3. During the registration process, the user sends the following details to the CPS_Q server:

$$< Username, \{appname\}, \{interest\} > \tag{4.2}$$

For example, the user may be interested in using the Restaurant Finder application and also be interested in receiving the coupons for Italian cuisine, or may be interested in getting sale

notification for shoes and baby clothes along with the coupons. So the user's registration string would be:

<ABC,{Restaurant Finder},{coupon_shoes, coupon_babyclothes, sale_shoes,sale_babyclothes}>.

The **User Information separator** module sends the Username to the **Unique code generator** module. This module generates a unique id, ID that is sent to the user. Along with the ID, the CPS_Q server also sends a subset of the Qcode database in the CPS_Q server consisting of the codes (q_{code}) for possible original queries of the applications that the user intends to use. The user may specify multiple applications he wishes to use and also multiple interests associated with receiving ads, coupons and sale notifications. The **User Information separator** module is a parser that parses this information, filters and stores it in different databases viz: User_apps db and User_Interests db, associating with each record a ID.

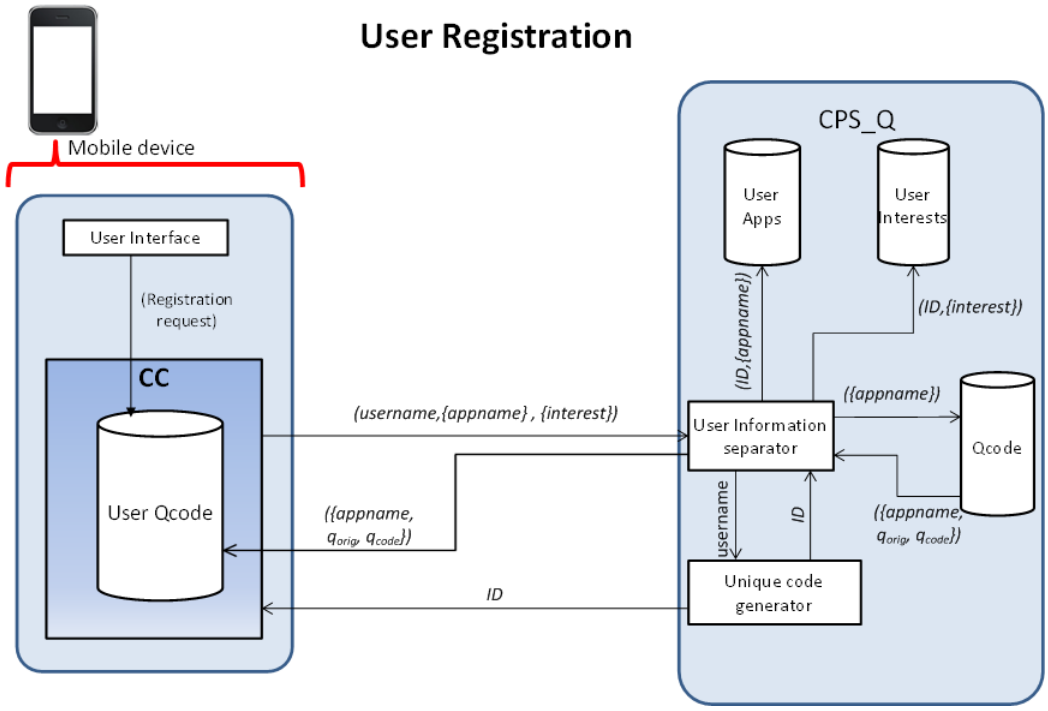


Figure 4.3: User Registration process

The updating of the User_Information (encrypted form), User Apps db and User Interests db is event triggered i.e. the user will update this information as and when there is a change (add / delete) application and interests of the user. The sequence diagram showing the details is shown in Figure 4.4. The sequence diagram is interpreted in a manner as explained in the earlier section of LBSP registration.

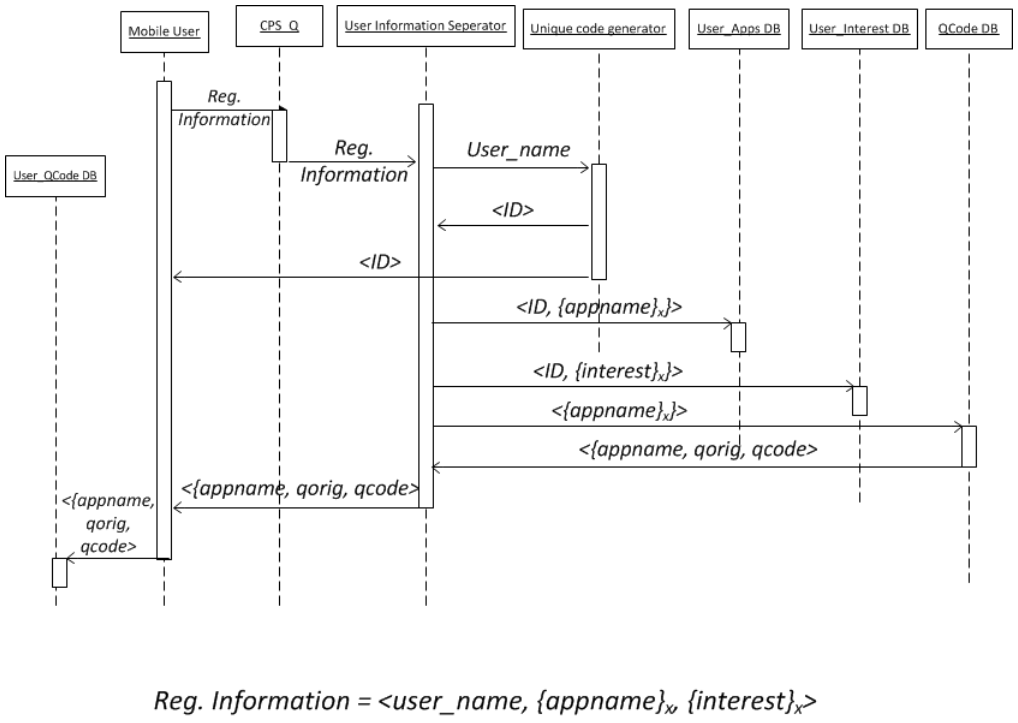


Figure 4.4: User Registration Sequence Diagram

4.3 Service Request Processing

The anonymization occurs in the service request processing activity. We propose novel methods of providing the *Location* privacy and the *Query* privacy while processing the request. We designed a *Split Cluster Anonymization Algorithm (SCAA)* that is used to anonymize the location and the query. Note the use of only id's, the unique request reference pair $< ID_m, ID_q >$, generated uniquely for each query, provides the *Identity* privacy too. Figure 4.5 depicts the communication between various components of the CLOPRO during the service request processing activity.

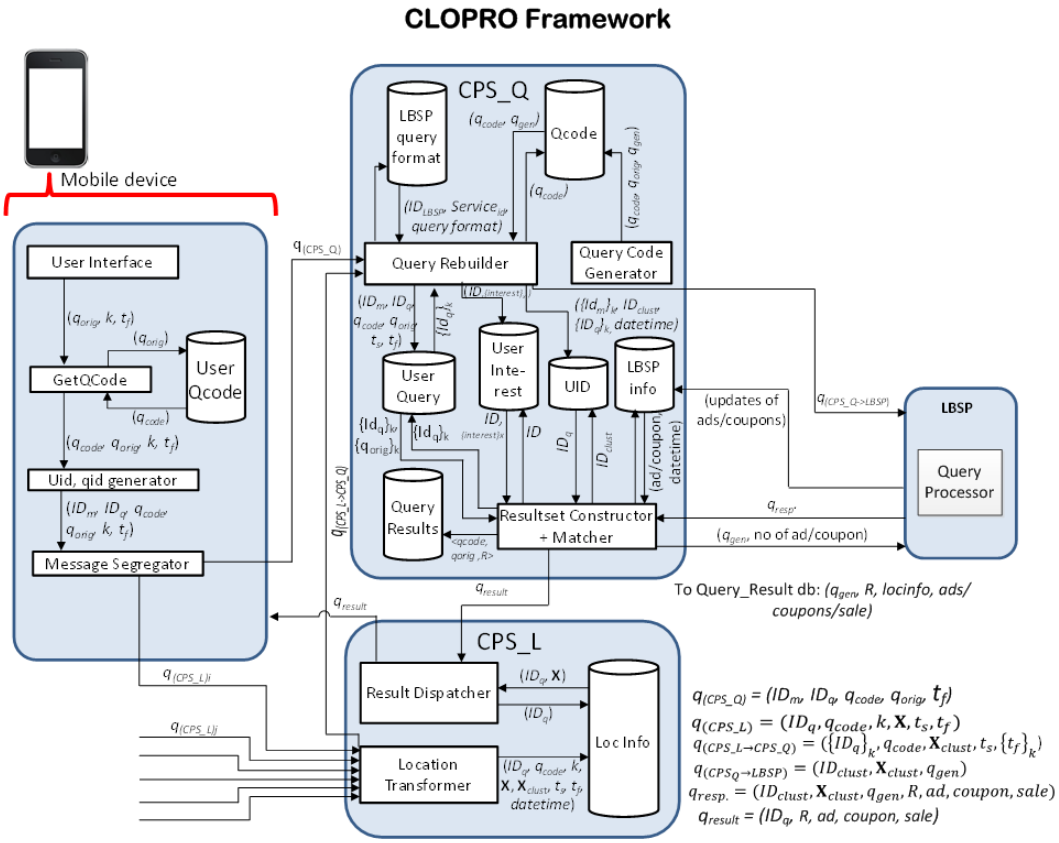


Figure 4.5: Service Request Processing

Figure 4.6 shows the sequence diagram of the service request processing showing the various entities involved in the service request processing and the movement of the message data.

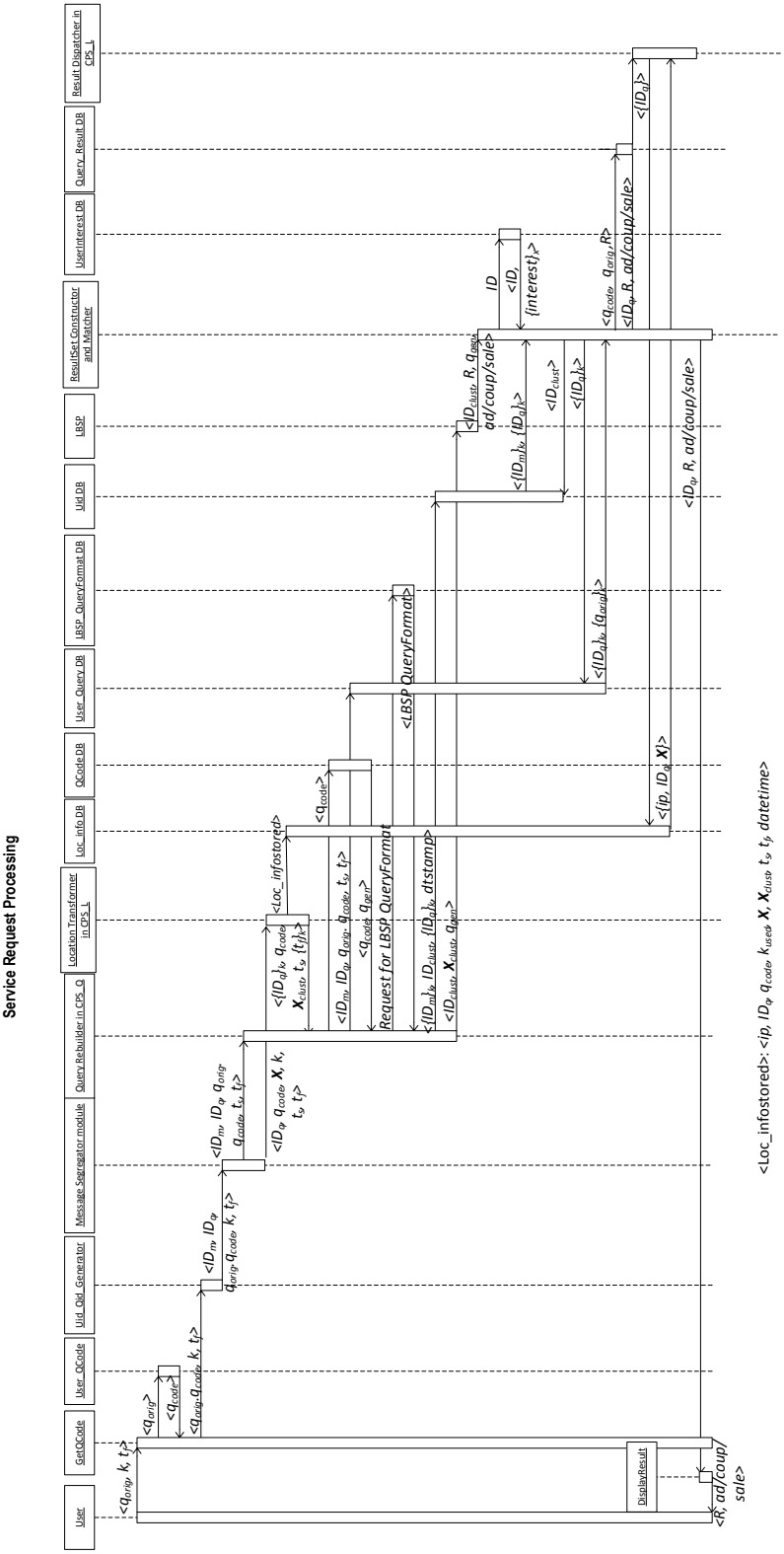


Figure 4.6: Service Request Processing Sequence Diagram

4.3.1 Operations on mobile device

The application installs a Context Cloaker (CC) module which handles all the communication between the user and the CPS_Q and CPS_L servers. When a user wishes to obtain a service from the LBSP, (s)he initiates a service request. Through the interface, the user specifies the service request $< appname, q_{orig}, k, t_f >$, where the desired level of anonymity is 'k' and the temporal resolution is (t_f), i.e. desired query response time. Depending on the application, additional information will be collected (such as *cuisine* or *type* of restaurant in the Restaurant Finder application) through the interface and appended to the query as a component of q_{orig_i}. So the final message that the information from the user interface (UI) sends to the CC module is of the form:

$$q_{UI \rightarrow CC} :=< appname, q_{orig_i}, k, t_f > \tag{4.3}$$

The CC module consists of three sub-modules:

- ♢ **Get QCode:** This module is responsible for fetching the code (q_{code_i}) associated with q_{orig_i} and then appends it to the query.
- ♢ **Uid_Qid_generator:** To guard against the possibility of identifying the user or to be able to match the identity of the user obtained by the adversary for the continuous queries, we create a new unique user identification and query identification, i.e. unique request reference pair $< ID_m, ID_q >$ every time the user initiates a service request to a LBSP. This module generates unique id for the user and the query. A simple mechanism is used for generating these is as follows:

$$ID_m := f(ID) \tag{4.4}$$

$$ID_q := f(ID) \tag{4.5}$$

More specifically,

$$ID_m := concat(ID + \text{'0'} + rand(0, 100000)) \tag{4.6}$$

$$ID_q := concat(user\_id + \text{'1'} + \text{rand\_number generated for } ID_q) \tag{4.7}$$

For example if the ID is 'ABC' then the ID_m may be 'ABC0123' and the respective ID_q will be

'ABC1123'. The CC module keeps a track of these id's to ensure that there is no duplication. We utilize the strong security properties of the Triple Data Encryption Standard (Triple DES) technique to encrypt these IDs. If an adversary is able to sniff the message, identifying the user or matching any two queries from the same user is not possible. The **Uid Qid generator** module then sends the complete query as defined below:

$$sr :=< ID_m, ID_q, q_{orig_i}, q_{code_i}, t_f > \tag{4.8}$$

♢ **Message segregator:** Once the components necessary for creating the queries, using simple system calls obtains the mobile device's lat-long ($\mathbf{X}_i$) information. It also obtains the system time and appends to the message. So the complete message is now of the form as shown in eq. (3.1)

$$sr :=< ID_m, ID_q, q_{orig_i}, q_{code_i}, k, \mathbf{X}, t_s, t_f >$$

This module splits the query into two components: one that is communicated to CPS_Q server and the other to CPS_L server as defined in the equations (3.2) and (3.3).

4.3.2 Operations by the CPS_L server

CPS_L server is responsible for providing location anonymity. We assume that at any point in time there will be many users who will have similar service requests which is a realistic assumption since the number of users using LBSP services is increasing exponentially. The two operations that are performed by the CPS_L server are *Location transformation* (Location Transformer module) and *Dispatching of the Results* and ads/coupons/sale notifications (Result Dispatcher module).

♢ **Location transformation process:** The Location Transformer module is responsible for clustering the many user service requests that arrive at the CPS_L server within a time window having the same query. Since multiple service requests are represented by one location coordinate, that of the centroid of the cluster, we can say that the users obtain the desired *k* anonymity.

A simple FIFO queue data structure $(Q_{q_{code}})_i$ stores the queries received by CPS_L server in the temporal order they are received for each q_{code}. An index is maintained for each live query (not dropped due expiration of the temporal requirement) from the set of queries not yet anonymized. The process is depicted in the *Location Anonymization Algorithm (LAA)* shown

in 4.1.

From each of the $(Q_{q_{code}})$ queue, all the queries obtained between the time interval $t+\delta t$ (where $\delta t \ll t_f$) and having the same q_{code} are sent to another list (L_q), which is in XML format. These are provided as an input to the Anonymization process. Note that the time constraint $\delta t \ll t_f$ is necessary for obtaining the results in desired query response times. The algorithm assigns the coordinates of the first element in list L_q to the centroid. Until the end of the list is reached, distance of each next element is tested and if the distance is more than a specific distance d, a new cluster is created. If not, the next query is added to the cluster and a new centroid is computed. The process continues until all the queries in the list are assigned to a cluster. The query centroid coordinate is assigned $\mathbf{X}_{clust}$.

This is followed by anonymization process where the maximum of k of all the users in each cluster is determined. If number of queries in a cluster is less than k_{max} , dummy queries with a dummy ID_q are generated such that it satisfies the required / specified k of every users' query in the cluster. The Loc_info db is updated and the query string $q_{CPS_L \rightarrow CPS_Q} := < \{ID_q\}_k, q_{code}, \mathbf{X}_{clust}, \{t_f\}_k >$ as defined in eq. (3.4) is sent to the CPS_Q.

♢ **Result Dispatch:** The second task performed by the CPS_L is to dispatch the results. This is a very simple operation with the Result_Dispatcher module fetching the exact location of the user who has issued a each query from the Loc_info db and dispatching the results to the specific user based on the ip address of the mobile user.

4.3.3 Operations by the CPS_Q server

The CPS_Q server performs two types of operations : online (Registration and Service Request Processing) and offline (Query Code Generation)

♢ **Registration of the Mobile Device User and the LBSP**: The CPS_Q server is responsible for registering the user and the LBSP. Upon receiving the registration data from the user, the CPS_Q server generates a unique id (ID) and sends it to the user along with a subset of the database comprising of the query codes (q_{code}) for the generic queries associated with applications of the user's choice. The CPS_Q server stores the details of the

LBSP upon request of registration, generates a unique identification for the LBSP and sends it to the LBSP.

♢ **Service Request Processing:** As seen in Figure 4.5, CPS_Q server receives inputs from two sources: the Mobile Device and the CPS_L server. The CPS_Q server checks if the result-set for the query exists in the database which stores the de-anonymized results (cached results). If it exists, the CPS_Q server returns the result-set to the CPS_L server. Otherwise, the query is rebuilt by the CPS_Q server by associating a unique id for the cluster (ID_{clust}) and replacing the query code with the associated generic query (q_{gen_i}). The process of generation of of generic queries is explained later. The CPS_Q server builds the request as per format requirement of the LBSP (Query_Rebuilder module). Each transformed request $sr^{'} \in SR^{'}$ eq. 3.9 is of the form:

$$sr^{'} :=< ID_{clust}, \mathbf{X}_{clust}, q_{gen_i} >$$

The service request sent to the LBSP will look like:
$< QWE, 38.0627, -85.37403, \text{“}Cuisine = Italian\text{”} and Type = \text{“}Fastfood\text{”} >$. Here, QWE is the cluster ID, the centroid coordinates are 38.0627,-85.37403, as calculated earlier, and the generic query for R22 is sent to the LBSP.

♢ **Communication to CPS_L server:** Upon receipt of result from LBSP, CPS_Q server parses the results and reconstructs request response for each user based on the original query (q_{orig_i}). The Resultset Constructor and Matcher module identifies the actual query of the user based on the q_{orig_i} by mapping the ID_q s' and also fetches the ID_m s'. The algorithm filters the ads, coupons, sale notifications to match the users interests and send the response to CPS_L server. Note that the CPS_Q server does not know the location of the user and hence this step is necessary. The server also caches these results for improving performance in case the query is repeated.

♢ **Query Code Generation:** The CPS_Q server performs this task offline, there by reducing the time required for the task of query aggregation. Each type of query is assumed to have certain features. For example, the Restaurant Finder application would have features regarding restaurants like the cuisine offered (F_1), the type of sitting arrangement (F_2), with or without bar (F_3) etc. These would be the features of the query. The name of the restaurant in this case will be leaf feature.The *Query Anonymization Algorithm (QAA)* 4.1 of the *SCAA* is used for Query abstraction by the CPS_Q server. The $F_{selected}$

is a set of combination of different features. In the Restaurant finder example, $F_{selected}$ will consists of $\{\{F_1\},\{F_2\},\{F_3\},\{F_1,F_2\},\{F_1,F_3\},\{F_2,F_3\}\}$. A generic query (q_{gen}) associating features in each subset is created and a code (q_{code}) is assigned to each generic query.

The number of POIs returned by the service provider as a result will definitely be more than the accurate results that can be obtained with the privacy protection mechanism. But this is an unavoidable trade-off mentioned earlier for the privacy protection. The CPS_L then sends the result back to the appropriate users location. The sequence diagram for service request processing is shown in Figure 4.5. If we cannot generate an appropriate cloaking region within the predefined period (t_f), we notify the user that the anonymization process failed. The system may request the user to change parameters *k,* t_f... to appropriate values.

4.4 Anonymization Algorithm

In this section we present the Split Clustering Anonymization (SCA) Algorithm for location anonymization and query abstraction. The complexity of Query Anonymization algorithm is $O(n^3)$. However, the Query Anonymization process is performed offline and so does not affect the time of operation of service request processing. The complexity of the Location Anonymization process id $O(n)$. This is a very simple implementation of the process of clustering. Also, the process ensures that the $< k, T >$ and the $< k, s >$ privacy protection. Since the k anonymity requirement of all the users is satisfied which may be different for each user, one can say that this algorithm also allows for personalized anonymity requirement.

Algorithm 4.1 Split Clustering Anonymization Algorithms

Location Anonymization Algorithm

Input: List of service requests SR initiated between $t_s$and $t_s + \delta t$ with same q_{code}
Algorithm:
$Cluster_1 \leftarrow sr_1$
$(\mathbf{X}_{clust_1}) \leftarrow (\mathbf{X}_1)$
for $2 \leq i \leq ||SR||$ **do**
  if $dist((\mathbf{X}_i),\ (\mathbf{X}_{clust_1})) < d$ **then**
    Add $(\mathbf{X}_i)$ to $Cluster_i$
    Compute new $(\mathbf{X}_{clust_i})$
  else
    Create new cluster
  end if
end for
for $\forall Cluster_i|$ **do**
  if $max(k) > no.of service requests in Cluster_i$ **then**
    Generate (max(k) - no. of service requests) in specified area
    Compute new centroid of all requests in each cluster
  end if
end for
Output: List of clusters with their centroids and the coordinates of original queries

---

Query Anonymization (Abstraction) Algorithm

Input:Set of $\{q_{orig}\} \in \mathcal{A}$, where $q_{orig} :< F_1, F_2, ..., F_n >$ where $F_n$leaf feature.
Algorithm:
for set $\{q_{orig}\} \in \mathcal{A}$ **do**
  Create set of Features used for clustering s.t.$< F_{selected} > \subset < F_1, F_2, ..., F_{n-1} >$
  Cluster $q_{gen} \leftarrow < F_{selected} >$
  Assign $q_{code} \leftarrow q_{gen}$
end for
Output: $\forall \mathcal{A},$ set of $\{q_{gen}, q_{code}\}$

CHAPTER 5

HEURISTIC ANALYSIS AND EVALUATION OF ATTACK MODELS

Once the model has been formalized, we first heuristically evaluate the success of the model by demonstrating, that the privacy of the user is preserved under various adversary attacks.

5.1 Adversary and Attack Models

Main goal of this model is to ensure that, at no stage during the entire query process, the link between the user's identity, the location and the time of initiation of the user query and the user's original query be known to the adversary. We say that privacy is breached if an adversary is able to establish this link.

Assumptions: The CLOPRO system servers (CPS_Q and CPS_L) are non-colluding. We also assume that the LBSP may also be an adversary and so the privacy parameters must be protected from LBSP too.

We do not consider that this framework can protect the user against the "global observer" like the "government" which has access to all the data.
To analyze this framework, we comprehensively analyze both the aspects: the attacks on the message strings during the communication and the attacks on the databases when the server is compromised. First we identify the information available at each of the servers. The information stored in the CPS_Q server in the various databases is given below:
• The ID , the original query (q_{orig}) , $< ID_m, ID_q >$ are in encrypted form, query code (q_{code}) for the generic query (q_{gen}), time of the initiation of the query (t_s) and the desired query response time (t_f).
• ID and *user_interests*
• ID and *appname*.

- ID and the original query asked by the user (q_{orig})
- Generic queries (q_{gen}) for original queries (q_{orig}) for each application.

The CPS_Q server does not have the information regarding following parameters:

♢ Exact location information of the querying users since the centroid coordinates calculated by the CPS_L server are the only coordinates known to the CPS_Q server.

♢ Only the ID is known, not the details of the user since the user name is not stored and used only once to create the unique ID. The other information of the user is stored in encrypted form.

The information stored in the CPS_L server:

♢ The ID_q, desired k, time at which the query was initiated t_s and the location information (**X**) along with desired query response time t_f.

The CPS_L does not have any information regarding the following:

♢ Issuer of the query (ID is not known).

♢ The original query (q_{orig}).

Based on the above discussion it can be observed that in this framework, no single entity other than the user has all the desired information.

♢ The CPS_Q does not know about the location of the user's who is issuing a query.

♢ The CPS_L does not know the user who is issuing the query or the original query of the user.

♢ The LBSP does not know any of the following: the user's id, his location or the actual query the issued by the user.

We assume that there two common types of attacks that are possible:

1. **Communication Attacks**: When the adversary sniffs the packets, man in the middle attack or snap shot attack and the replay attack.

2. **Database Attacks:** When any one of the server is compromised and the adversary has full access to all the databases on that server.

In all the cases, the entire communication is done using identities and a unique ID_m and ID_q will be generated for each of queries so the adversary will not be able to determine the identity of the user at any time.

5.2 Solutions

5.2.1 Communication attacks

5.2.1.1 Man in the Middle Attack or Snap shot attack:

During the process of obtaining the LBSs, there are various message strings used for communication. If these message strings are sniffed by an adversary, they can reveal the links between the various privacy parameters that can breach the privacy of the user when the adversary is able to get access to them. There are four entities in the system: the Mobile Device User, CPS_L server, the CPS_Q server and the LBSP. In the following section, we analyze each of the communication message strings that are exchanged between the various entities in the system that an adversary can get access to. The service request message strings that pass between various components of the CLOPRO are depicted in Figure 5.1. The messages shown from top to bottom are temporally sequenced.

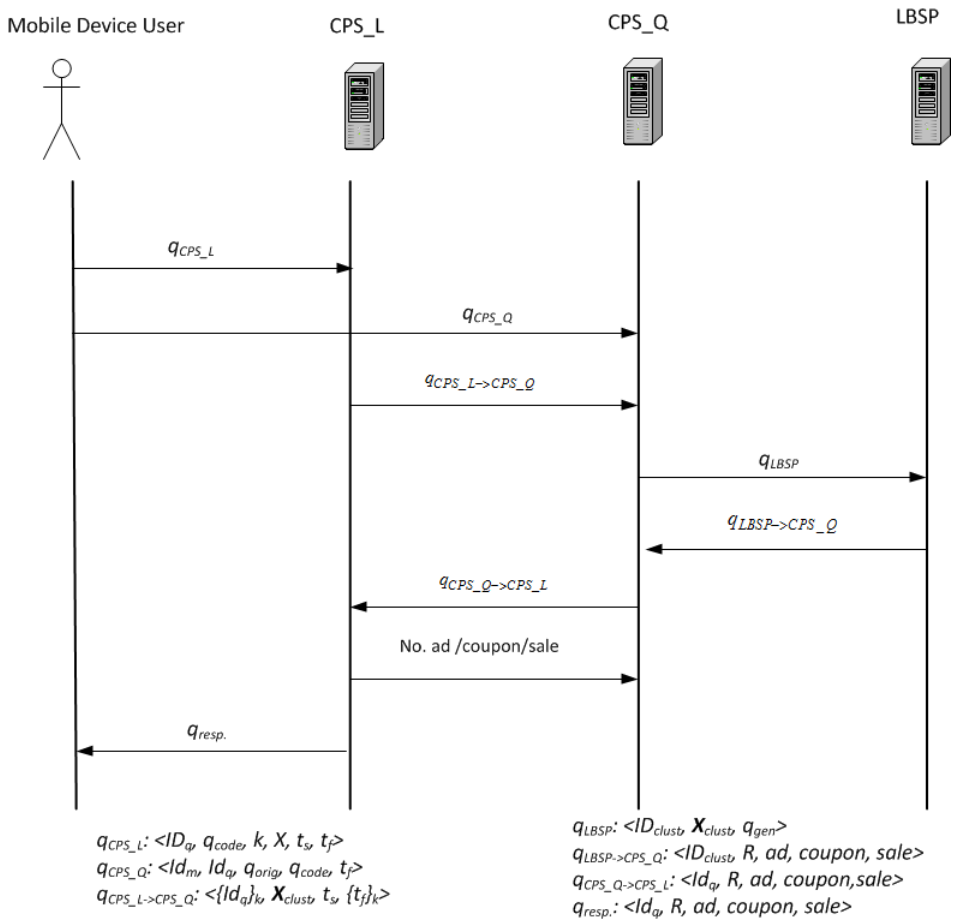


Figure 5.1: Service Requests between various components of CLOPRO

We consider the adversary of type honest but curious adversary (follows the protocol diligently; however they may try to obtain more information than needed by providing forged input data) [11]. We now analyze each of the message strings shown in the figure 5.1 that the adversary is likely to sniff for identifying the mobile user details:

$q_{CPSL} := \ < ID_q, q_{code}, k, \mathbf{X}, t_s, t_f >$: The user's id ID and the original query q_{orig} are not a part of this query string. If the sequence of messages are sniffed by the adversary, the only information that the adversary will be able to infer is the sequence of ID_qs' (in encrypted format) and so cannot be linked to any particular user, and the time of issue of query. However, if the user is not continuously querying, the adversary cannot establish the linkage between the various privacy parameters. The adversary can infer the desired anonymity level that is not of much use since the queries that are clustered are not known ahead of time and so it is impossible to establish the linkage or any associated information regarding the user.

$q_{CPSQ} := < ID_m, ID_q, q_{code}, q_{orig}, t_f >$: The location coordinates of the user are not a part of this query string and hence the adversary cannot establish a link between the user and the location of the user. Even, if the adversary gets a history of messages, the adversary cannot, on scanning the messages, know that a particular user since the ID_m and ID_q are encrypted before being sent to the CPS_L or the CPS_Q server. The adversary can only know that there is a specific query code associated with a query.

$q_{CPSL \rightarrow CPSQ} := < \{ID_q\}_k, q_{code}, \mathbf{X}_{clust}, t_s, \{t_f\}_k >$: The adversary can infer the location coordinates of *k* users (may be real or fake) and they have asked same query at that instant of time. The original query is not known and so with this partial information the adversary cannot identify any other privacy threatening information about the user or the query neither the link.

$q_{LBSP} := < ID_{clust}, \mathbf{X}_{clust}, q_{gen} >$: Through the knowledge of this snap shot of the query, the real user's id (this is the ID_{clust} generated for *k* user's) or the actual user query is not disclosed to the adversary. The adversary can decipher that this query has been asked by the users from this location or the area (MBR) depending on whether the centroid coordinates or the MBR was part of the query sent to LBSP.

$q_{LBSP \rightarrow CPSQ} := < ID_{clust}, q_{gen}, \{R\}, \{ad/coupon/sale\} >$: As can be seen from the parameters of the message string, the adversary can at most associate the generic query and the ads/ coupons that are associated at that instant of time. This information is available with the LBSP. However, the relation between its association with the query or the location is not revealed.

$q_{resp.} := < \{ID_q, R, ad, coupon, sale\} >$: The adversary may be able to guess the answer to the query. If a sequence of messages are sniffed, the adversary may learn about the ads/ coupons that a particular with a specific id has received.

5.2.1.2 Replay attacks

We now consider the cases when the adversary spoofs a number of message strings for the CPS_L server. In that case, the CPS_L server can decipher that these queries are fake since there cannot be multiple queries from the same user with the same time stamp since the ID_q has the unique ID appended which is used to determine the user. The CPS_L server will then drop these query strings or mark them as spoofed and not include them in their clustering process.

If the adversary creates fake queries in communication to the CPS_Q server (based on the message going to the CPS_Q server, the adversary has not sent the location information. Since the part of query is processed by CPS_L server, this piece of information will be missing to process the query (other part of message has not gone to the CPS_L server and so the CPS_L server does not send the $(\mathbf{X}_{clust})$ for those ID_q*'s*). It can thus be identified that the query is fake and no results will be sent to the adversary.

Thus, the model is secure against this setting.

5.2.2 Attack on databases through the access to server

We now analyze the situation that any one of the servers is attacked. If any one of the servers is compromised, the adversary gets access to all databases of that servers:

CPS_Q server has following databases:

- ♢ Qcode db: $(q_{orig}, q_{gen}, q_{code})$
- ♢ User_Query db: $(ID_m, ID_q, q_{code}, q_{orig}, t_f)$
- ♢ LBSP_QueryFormat db: *(Service_id, Queryformat)*
- ♢ Uid db: ($ID_{mk}, ID_{clust}, ID_{q_k}, datetime$)
- ♢ User_Interests db: (ID,$interest_{i_x}$)
- ♢ User_Apps db: (ID,$appname_{i_x}$)
- ♢ LBSP_Services db: *(Service_id, ServiceName, ServiceDescription)*

♢ LBSP_Info db:

$(Service\_id, ad, loc_{ad}, validity_{ad}, coupon, loc_{coup}, validity_{coup}, sale, loc_{sale}, validity_{sale})$

CPS_L server has following databases:

♢ Loc_info DB: $(ID_q,\ q_{code},\ k,\ \mathbf{X},\ \mathbf{X}_{clust},\ t_s,\ t_f,\ datetime)$

As seen from the details of the databases in the CPS_Q or the CPS_L server, if one of the servers is compromised the adversary cannot get complete information about the user, the original query and the location from where the query was initiated.

Thus, the model is secure against this setting.

Thus we have heuristically shown that the CLOPRO system preserves the privacy of the user while using the services provided by LBSP.

CHAPTER 6

CLOPRO FRAMEWORK ANONYMIZATION CHARACTERIZATION AND EVALUATION

In this chapter we present a formal treatment to the problem. We first define the terms and definitions used in our framework and further propose a definition for the Privacy Protection. We use Lemmas to prove our proposition and demonstrate successfully that CLOPRO is able to successfully provide the requisite privacy to the user while protecting the revenue model of the the LBSP by delivering the *CAPPA* to the user. The CLOPRO framework achieves the objective of privacy protection using the following three transformations:

♢ **Identity Protection** (T_Rid)

♢ **Query Anonymization (Abstraction)** (T_Rquery)

♢ **Location Anonymization** (T_Rloc)

6.1 Terms and Definitions

In this section we present some terms and definitions used to formally define our problem and prove that the privacy is preserved using our framework while protecting the revenue model of the LBSP. So, we have a two fold objective that our system must handle

♢ Protect the privacy of the user by ensuring that the service provider is unable to link the user's identity with the location and time of query.

♢ The Service provider must be ensured revenue from the A & A modules through the mechanism of *Context Aware Advertising* to the user, even if it does not have the information required for profiling the user.

Anonymization Set: The Anonymization set is defined as the list of probable users requesting for various services from the LBSP.

Formal definition of Attack: It is assumed that the adversary has the following knowledge: It is assumed that the adversary has the knowledge of the transformed service request ($sr^{'} \in SR^{'}$) received by the LBSP

$$sr^{'} :=< ID_{clust}, \mathbf{X}_{clust}, q_{gen_i} >$$

The adversary may also be aware of k , the number of users in a cluster. The attack is said to be successful, when based on the given knowledge, the adversary can infer the contents of original request

$$sr_i :=< (ID_m, ID_q), q_{orig_i}, q_{code_i}, k, \mathbf{X_i}, t_s, t_f >$$

Query Abstraction: Query Abstracter creates a layer of query abstraction by modifying the content of the query in the service request by modifying the granularity level of the requested service. This is achieved by organizing all the queries in a tree structure where each node represents the abstracted query for all the services in the subtree rooted at that node.

Revenue model of the LBSP: It is well known that the LBSP's revenue is tied to the number of advertisements, coupons and sale notifications the LBSP is able to deliver to users. The revenue is generally more if the deliveries are relevant i.e. context aware. So we say the **revenue model of the LBSP is protected** if the user receives the **relevant** ads / coupons and sale notifications.

6.2 Phased transformation of the service request

The CLOPRO framework achieves the objective of privacy protection using the following three transformations:

1. Identity Protection (T_Rid): This is achieved through the generation of a unique request reference pair $< ID_m, ID_q >$ for each query and encrypted on the mobile device to protect the identity of the user. Also, a unique ID_{clust} is generated for k users in a cluster when presented to LBSP. Thus, we can say $T_Rid : u^{'} = f(u)$ where f is the identity transformation function.

2. Query Anonymization (Abstraction) (T_Rquery): This is achieved through the replace-

ment of original query q_{orig_i} by a generic query q_{gen}. The creation of the generic query is handled by the CPS_Q server offline. One can say that a single query to the LBSP comprises of a cluster of the actual queries from a set of users. The query anonymization transformation is

$$T_R query: \ q_{gen_i} \quad := \quad \oplus_{i=1}^{n_i-1} a_i F_i \text{ where } \begin{cases} a_i = 0 & F_i \text{ is } absent \\ a_i = 1 & F_i \text{ is } present \end{cases}$$

3. Location Anonymization ($T_R loc$): This is achieved through the process of clustering several service requests arriving at Location Anonymization Server (CPS_L) within a time window for service requests having same generic query. The transformation is of the form $T_R loc : C_i^{'} = g(C_i)$ where g is the transformation function and C_i and $C_i^{'}$ represent the context of the user and i represents the number of contexts. In this model, $C_i = (C_1, C_2)$ where C_1 is the location and C_2 is the time of initiation of query.

The complete service request, after modification by the resident application on the mobile device, is as shown in eq.(3.1):

$$sr_i :=< (ID_m, ID_q), q_{orig_i}, q_{code_i}, k, \mathbf{X_i}, t_s, t_f >$$

Thus, the definition of this set of service requests can be formally represented as follows:

$$SR \in U \times \prod_j C_j \times \prod_i q_{orig_i} \times t_f \tag{6.1}$$

Here U is the unique request reference pair $< ID_m, ID_q >$ generated for every service request; $C_j = (c_1, c_{2,}.....c_n)$ is the context of the service requester (desired anonymity,location and time of query), the desired anonymity is denoted by k, location coordinates of the user are denoted by $\mathbf{X}$ and the request initiation time or start time of request is t_s (For eg: c_1 may be the location of the user, c_2 may be the start time of service) ; $q_{orig_i} = (q_{orig_1}, q_{orig_2}, ...q_{orig_n})$ is the actual query of the user and the desired service response time is denoted by t_f. If the service response is not obtained within that specified period, the request is dropped and a message is issued to the user that the processing time for the request to be reissued needs to be increased. and the request should be reissued i.e. t_f can be seen as a deadline for the request.

When the transformations are applied, the transformed request consists of the following as shown in eq. (3.9) (We will consider just one form here. The same logic can be applied to other format also).

$$sr^{'} :=< ID_{clust}, \mathbf{X}_{clust}, q_{gen_i} >$$

where ID_{clust} is the *id* associated with each cluster of k service requests (real or dummy); $(\mathbf{X}_{clust})$ consists of transformed location coordinates and q_{gen_i} is the transformed query. Thus we can say,

$$SR^{'} \in U^{'} \times \prod_{j} C_j^{'} \times \prod_{i} q_{gen_i} \tag{6.2}$$

where

$$U^{'} := f\,(U) \tag{6.3}$$

$$C_j^{'} := g\,(C_j) \tag{6.4}$$

and

$$q_{gen} := \oplus_{i=1}^{n_i - 1} a_i F_i \text{ where } \begin{cases} a_i = 0 & F_i \text{ is } absent \\ a_i = 1 & F_i \text{ is } present \end{cases} \tag{6.5}$$

Note: f is quid transformation, g is location transformation and F_i is set of features of original query: $q_{orig}\colon\ =< F_1, F_2, ..., F_n >$

The transformation is achieved using the transform function T_R by mapping the service request sr to $sr^{'}$ as shown in eq. (6.6)

$$T_R\colon\ sr^{'} = T_R\,(sr) \tag{6.6}$$

Thus, a **successful attack** can be defined as:
If an adversary knows $sr^{'} \in SR^{'}$ such that $SR^{'} \in U^{'} \times \prod_{j} C_j^{'} \times \prod_{i} q_{gen_i}$
and has been obtained using the transformation function
$T_R :< T_R id \cup T_R query \cup T_R loc >$
an adversary is able to decipher

$sr \in SR$ such that $SR \in U \times \prod_j C_j \times \prod_i q_{orig_i} \times t_f$

For the transformed service request to be privacy protecting, through the use of transformation function $T_R \colon SR \rightarrow SR^{'}$, has to hold the following properties:

♢ **Many to One mapping** (Surjective): A function f is said to be surjective (or onto), or a surjection from a set X to a set Y, if every element $y \in Y$ has a corresponding element $x \in X$ given by $f(x) = y$. When applied to the Transformation function T_R, this property states that two or more service requests can be transformed into the same modified request when presented to LBSP.

$$\exists\,(sr_1, sr_2) \in SR, T_R\,(sr_1) = T_R\,(sr_2) = sr^{'}, (sr_1) \neq (sr_2) \tag{6.7}$$

♢ **Non Invertible:** If an input x into the function f produces an output y, then putting y into the inverse function g produces the output x, and vice versa. i.e., $f(x) = y$, and $g(y) = x$. A function f that has an inverse is called invertible. In other words, it is impossible to apply some inverse transformation T_R^{-1} to $sr^{'}$ to obtain sr

$$\forall T_R(sr) = sr^{'}, \nexists T_R^{-1} \colon T_R^{-1}(sr^{'}) = sr \tag{6.8}$$

♢ **Asymmetric:** The transformation function T_R and it's inverse T_R^{-1} are not symmetric in nature(since T_R^{-1} does not exist in the CLOPRO framework case).

$$\forall sr \in SR \colon T_R^{-1}(T_R\,(sr)) \neq sr \text{ where } T_R^{-1} \colon SR^{'} \rightarrow SR^n, n \geq 1 \tag{6.9}$$

♢ **Non injective:** An injective function is a function that preserves distinctness. It is sometimes called many to one function. The transformation function T_R is non injective in nature as formally specified below:

$$\forall sr_1, sr_2 \in SR \colon T_R(sr_1) = T_R(sr_2) \implies (sr_1) = (sr_2) \tag{6.10}$$

Gedik *et al* [7] have defined **Anonymity Requirement** as "$k - anonymity$ requirement demands that for each perturbed message, there exists at least$k$-1 other perturbed messages

with same spatio temporal cloaking box, each from a different mobile client." We extend this definition to include the query anonymization through abstraction as:

" A user, his query and the context in the service request must be indistinguishable from 'k-1' other user's queries and the user's location in their respective service request (identified by unique $< ID_m.ID_q >$ pair) and it should be impossible to link the user's identity, the user's context, the time of requesting the service and the original query."

To ensure that there is no breach of privacy, there are three components of the service request that need to be protected. In this context, the Privacy Protection is defined as:

Proposition 1:

Given that $sr :=< uid : u, Context : C, query : q >$ where $C = (c_1, c_{2,}..., c_n)$ and $sr^{'} :=< uid : u^{'}, Context : C^{'}, query : q^{'} >$
which is obtained on application of the transformation $T_R :=< T_Rid \cup T_Rquery \cup T_Rloc >$. The following is not possible $T_R^{-1}(sr^{'}) \rightarrow sr$. The revenue of the LBSP will also be protected.
We prove this proposition using the following four Lemmas:

1. The transformation function T_Rid is surjective, non invertible and non-injective in nature and so Identity of the user is not revealed in the CLOPRO framework. (Lemma 1)
2. The transformation function T_Rquery is surjective, non invertible and non-injective in nature and so the original query in the service request of a user is not revealed through use of CLOPRO framework. (Lemma 2)
3. The transformation function T_Rloc is surjective, non invertible and non-injective in nature and so the location of a specific user is not revealed through the use CLOPRO framework. (Lemma 3)
4. The user receives $CAPPA$ without the LBSP being aware of who receives the ads, where is the receiver located and what is the exact query of the user. (Lemma 4).

Lemma 1: *The transformation function T_Rid is surjective, non invertible and non-injective in nature and so Identity of the user is not revealed in the CLOPRO framework.*

Proof:

To uniquely identify a user's service request, a request reference pair is generated for every request. This implies that no two service requests, even from the same user, have the same $< ID_m, ID_q >$ pair making $T_R id$ surjective. The function is also non-injective An injective function (many to one function) is a function that preserves distinctness. The transformation function T_R is non injective in nature if $\forall sr_1, sr_2 \in SR$: $T_R(sr_1) = T_R(sr_2) \not\Longleftrightarrow (sr_1) = (sr_2)$). Also, since this unique request reference pair is encrypted using AES encryption algorithm at the user's device itself making it impossible to relate the origin of the service request to a particular user. Secondly, when the request is sent to the LBSP, a unique ID_{clust} is generated for a cluster of 'k' users making the transformation function non-invertible (If an input x into the function f produces an output y, then putting y into the inverse function g produces the output x, and vice versa. i.e., $f(x) = y$, and $g(y) = x$. A function f that has an inverse is called invertible. In our case, the *Query transformation* (abstraction) and *Location transformation* are non invertible since it is not possible to obtain original query or original location coordinates from the transformed values. It is not possible to obtain the original query from the abstracted query (you cannot figure out that Restaurant name is Pizza Hut from the query that says the Cuisine is 'Italian' or Type = 'Fastfood') or the original coordinates of the users' whose service requests are in a cluster from the centroid of that cluster respectively. In other words, it is impossible to apply some inverse transformation T_R^{-1} to $sr^{'}$ to obtain sr i.e $\forall T_R(sr) = sr^{'}, \nexists T_R^{-1}$: $T_R^{-1}(sr^{'}) = sr$). Since $T_R id$ satisfies the property desirous for the transformation function to achieve privacy protection the identity of the user is not revealed in the CLOPRO framework. Lemma 1 can be demonstrated diagrammatically using Figure 6.1.

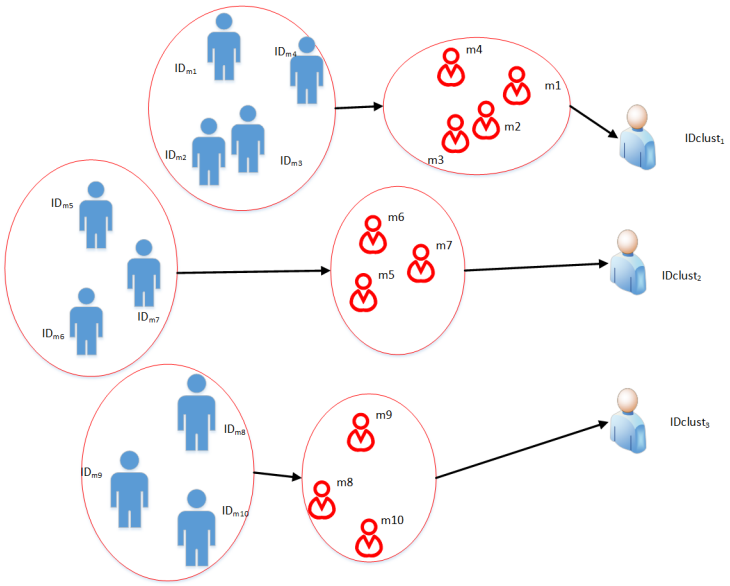


Figure 6.1: Privacy Protection through Identity Transformation

Lemma 2: *The transformation function* $T_R query$ *is surjective, non invertible and non-injective in nature and so the original query in the service request of a user is not revealed through use of CLOPRO framework.*

Proof:

The Query anonymization algorithm shows that the query transformation $T_R query$ that has "Many to one" property which implies that there can be more than one service requests that result in the same transformed query (Surjective). Since it is not possible to obtain the original query from the generic query it can be said that the transformation is non-invertible. Also, due to generalization of the query sent to the LBSP, the distinctness of the query is lost and hence is non-injective in nature and thus one can say that *Query* privacy is protected. Thus, we prove that original query in the service request of a user is not revealed through use of the CLOPRO framework. Lemma 2 can be demonstrated diagrammatically using Figure 6.2.

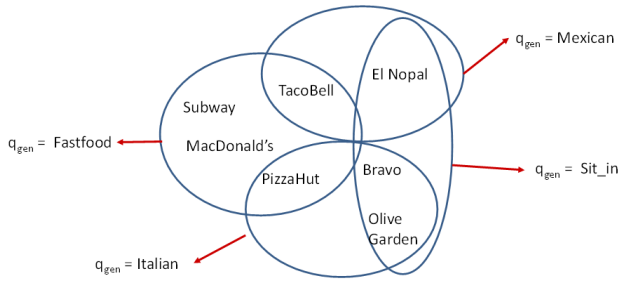


Figure 6.2: Privacy Protection through Query Transformation

Lemma 3: *The transformation function T_Rloc is surjective, non invertible and non-injective in nature and so the location of a specific user is not revealed through the use CLOPRO framework.*

Proof:

Location transformation T_Rloc is non-invertible since it is not possible to obtain the original location coordinates from the transformed values (centroid coordinates). Since an inverse to this transformation function does not exist, it follows that it is not possible to obtain the exact location coordinates of the user given the cluster centroid as location for 'k' users. Also, the transformation function follows the Surjective property since different transformations may result into same transformed result. It is impossible to find out the original coordinates of the service requester even if the adversary knows the centroid coordinates and the desired anonymity level, since the same centroid can be obtained from various sets of 'k' points. Also, different number of 'k ' service requests might result in same centroids for some different generic queries. The request initiation times of different users in the cluster of 'k' users fall in a time window and are not exactly the same for all the 'k' users in the cluster. Thus, it is proved that it is impossible to reveal the location coordinates and the request initiation time i.e. the context $(\mathbf{X}_i, t_s)$ of each individual user in the cluster. Lemma 3 can be demonstrated diagrammatically using Figure 6.3.

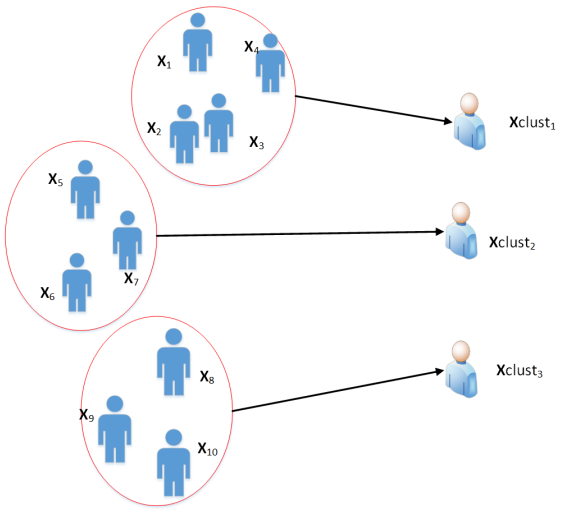


Figure 6.3: Privacy Protection through Location Transformation

Lemma 4: *The user receives the advertisements based on his / her context and interests without the LBSP being aware of the user or his (her) context.*

Proof:

In the CLOPRO framework, with every service request served by the LBSP, the CPS_Q also receives the ads/ coupons/ sale notifications relevant to the users request (coupons for 'Mexican' restaurants if generic query is to find locations of Mexican restaurants, along with other ads / coupons/ sale notifications of the stores in the vicinity of the $(\mathbf{X}_{clust})$ coordinates in the query. The **Matcher** module of the **Resultset Constructor and Matcher** module in CPS_Q matches the user's interest, user's original query and then filters out the relevant ads / coupons / sale notifications for the specific ID_q of the user. Thus, the model ensures that the ads / coupons / sale notifications forwarded to the user are relevant to the user's context., hence we call it as *Context Aware Privacy Protecting Advertising*. The LBS Provider, updates the ads/ coupons/ sale notifications to the CPS_Q from time to time. Thus, even when the CPS_Q is servicing the request through cached results of the service requests processed earlier, context aware ads/ coupons/ sale notifications can be served to the user. Information regarding the number of ads/ coupons/ sale notifications delivered to the user is sent to the service provider, who can then claim the revenue for serving the ads/ coupons/ sale notifications from A & A modules.

In the proposed transformation functions, inverse does not exist for any of the transformation functions. Hence all the transformation functions are Asymmetric in nature.

Proof of Proposition 1:

Lemma 1 proves protection of the *Identity* privacy, Lemma 2 proves the *Query* privacy, Lemma 3 proves the privacy of the location and time of the service request. With every service request, the advertisement / coupon / sale notification relevant to the user are delivered which is necessary for generation of revenue without revealing the user or his context (Lemma 4). Since privacy of each component of the service request is protected, the privacy of the linkage amongst these components is also preserved and hence it can be said that the privacy of the user is protected when the user chooses to use the CLOPRO framework.

CHAPTER 7

GRAPH - THEORETIC APPROACH TO EVALUATING PRIVACY PROTECTION

Graph theoretical ideas have been an important technique to represent and /or solve many problems in computer science, especially in research areas such as networking, data mining, clustering, image capturing and image segmentation. For example, a data structure can be represented in the form of a *tree* which in turn utilized vertices and edges. It has been used in many diverse areas like chemistry to show the construction of bonds or study of atomic structures, or representing and predicting the spread of the diseases in biology, or resource planning and scheduling of large and complicated projects (CPM/PERT) in Operations Research. Few other popular application has been for studying the graph coloring problem and traveling salesman problem. Since networks can be easily represented using graphs, it has been used to model network topologies and simulate many associated problems such as the propagation of stealth worms on large computer networks and design optimal strategies to protect the network against virus attacks in real time (Vertex Cover Algorithm), identifying connectivity issues in adhoc networks (graph spanners), or modeling the web, to name a few. A graph typically shows the connectedness between various vertices / parameters through the use of edges. In the current context, we represent the decipherability of various privacy parameters as a Directed Acyclic Graph (DAG). The edges of the graph indicate the decipherability relations between the parameters (vertices) based on the apriori knowledge of the adversary under different condition of attacks. The strongly connected components of a graph indicate the reachability between the vertices. We use this concept of strongly connected components to show that under all possible conditions, the privacy of the user is protected as the resulting output graphs have one or more privacy parameters that are not a part of strongly connected components of the graph.

7.1 Graph Theory Basics

Any network of parameters can be represented as a directed graph where the vertices represent the parameters and edges represent the decipherability of the parameters based on knowledge of some other parameters. This abstraction allows for implementing the graph based techniques and algorithms to analyze the graph. In this section, relevant basic notations and the definitions used in the graph theory are reviewed[62].

Graphs

A *graph G* is a finite, nonempty set *V* together with a (possibly empty) set *E* (disjoint from *V*) of two-element subsets of (distinct) elements *V.* Each element of *V* is referred to as a *vertex* and *V* itself as the *vertex set* of *G, V(G);* the members of the *edge set E (E(G)),* are called *edges.* An element of a graph can mean a vertex or an edge. An edge $e = (a, b)$ is said to join vertices *a* and *b*. The $e = (a, b)$ is an edge of a graph *G*, the *a* and *b* are *adjacent vertices* while *a* and *e* are *incident* as are *b* and *e.* Also, if there are two distinct edges e_1 and e_2 of *G* incident with a common vertex, then e_1 and e_2 are *adjacent* edges.

Directed Graphs and Networks

A *directed graph* G= *(V,E)* is a finite nonempty set V of vertices and a set E of edges whose elements are ordered pairs of distinct vertices of set *V*. Figure 7.1 shows an example of a directed graph. For this graph, V={1, 2, 3, 4, 5, 6, 7, 8} and E= {(1,2), (1,3), (2,5), (3,4), (4,2), (4,5), (4,8), (5,6), (5,7), (6,4) (6,8), (7,8)}. The number of elements in the set *V* is called the *order* of the graph and the cardinality of the set *E* is the *size* of the graph. A directed network is a directed graph whose vertices and / or edges have associated numerical values (typically costs, capacities, and /or supplies and demands). In the subsequent sections, the terms graphs and networks are used interchangeably.

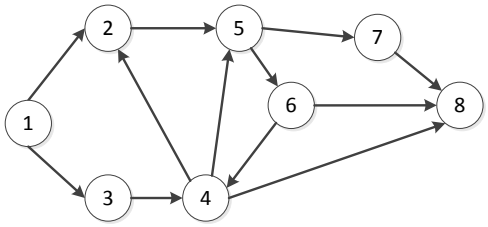


Figure 7.1: Directed Graph example

Undirected Graphs and Networks

An undirected graph is similar to the directed graph except that the edges are unordered pair of distinct vertices. Figure 7.2 shows an example of an undirected graph.

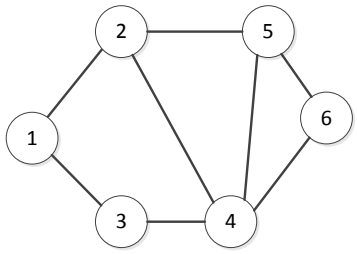


Figure 7.2: Undirected graph example

Tails and Heads

A directed edge (i, j) has two endpoints i and j. Vertex *i* is referred to as the *tail* of edge *(i ,j)* and vertex *j* is referred to as its head. An edge *(i,j)* is *incident* to vertices *i* and *j*. The edge *(i ,j)* is an *outgoing* edge of vertex *i* and an *incoming* edge of node *j*. Whenever the edge $(i, j) \in E$, vertex *j* is said to be *adjacent* to vertex i.

Degrees

The *degree of a vertex v* in a graph *G* is the number of edges of *G* incident with v. The *indegree* of a vertex is the number of incoming edges at v *(*number of edges *adjacent* **to** v*)*. and its *outdegree* is the number of the vertex v's outgoing edges (number of edges *adjacent* **from** v). The *degree* of a vertex is the sum of its indegree and outdegree. Thus, $deg\ v = od\ v + id\ v$. For example in Figure 7.1, vertex 4 has indegree of 2 (3->4, 5->4) and outdegree of 3 (4->2, 4->5, 4->8), and a degree of 5. It has been established that for a graph with *order x* and *size y* [62],

$$\sum_{i=1}^{p} od\ v_i = \sum_{i=1}^{p} id\ v_i = y \tag{7.1}$$

Adjacency list

The *edge adjacency list A(i)* of a vertex *i* is the set of edges emanating from that vertex, that is, $A(i) = \{(i, j) \in E : j \in V\}$. The *vertex adjacency list A(i)* is the set of vertices adjacent to that vertex, in this case, $A(i) = \{j \in V : (i, j) \in E\}$.

Subgraph

A graph $G^{'} = (V^{'},\ E^{'})$ is a *subgraph* of $G = (V, E)$ if $V^{'} \subseteq V$ and $E^{'} \subseteq E$. $G^{'} = (V^{'}, E^{'})$ is said to be a subgraph of $G = (V, E)$ induced by $V^{'}$if $E^{'}$contains each edge of $\boldsymbol{E}$ with both endpoints in $V^{'}$. The simplest subgraph can be obtained by the deletion of a vertex or edge. This A graph $G^{'} = (V^{'}, E^{'})$ is a *spanning subgraph* of $G = (V, E)$ if $V^{'} = V$ and $E^{'} \subseteq E$.

Walk

A *walk* in a directed graph $G = (V, E)$ is a subgraph of G consisting of a sequence of vertices and edges $i_1 - a_1 - i_2 - a_2....i_{r-1} - a_{r-1} - i_r$ satisfying the property that $\forall 1 \leq k \leq r - 1$, either $a_k = (i_k, i_{k+1}) \in E$.or $a_k = (i_{k+1}, i_k) \in E$.. The graphs in Figure 7.3 (a) (Walk 1-8: 1-2-1-3-4-8), Figure 7.3 (b) (Walk 1-2: 1-3-4-2) illustrate the two walks in this graph. An *open walk* is when the initial vertex and the ending vertices are not the same (Figure 7.3 (b)) and the *closed walk* is when the initial vertex and the ending vertices are the same (eg: Walk 1-2-4-3-4-2-1). A *trivial walk* is the one that does not contain any edges (Walk 1 or Walk 2).

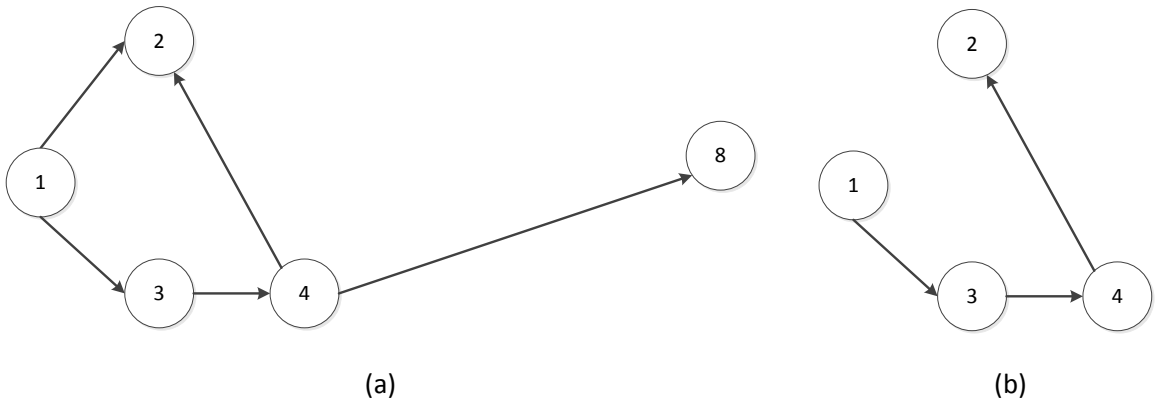


Figure 7.3: Examples of walks

Directed walk

A *directed walk* is a walk with an "orientation". *In a directed walk,* any two consecutive vertices i_k and i_{k+1} on the walk, $(i_k, i_{k+1}) \in E$. The walk shown in Figure 7.3(a) is not directed, while walk in Figure 7.3(b) is directed.

Path

A *path* is a *walk* where none of the vertices are repeated. The *walk* shown in Figure 7.3(b) is also a *path*, but the *walk* shown in Figure 7.3(a) is not because it repeats vertex (2) twice.

Directed path

A *directed path* is a path in a directed walk without any repetition of vertices. Figure 7.4 shows an example of directed path 1-2-5-7-8.

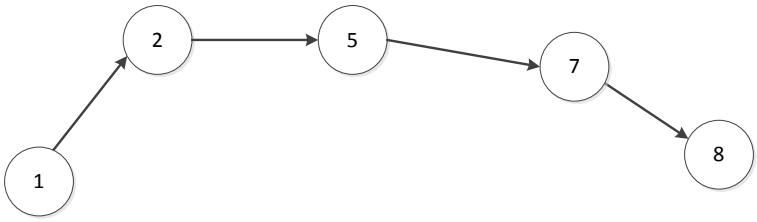


Figure 7.4: Example of Directed Path

Cycle

A *cycle* is a path $i_1 - i_2 - ....i_{r-1} - i_r$ together with the edge (i_1, i_r) or (i_r, i_1). A cycle is generally referred using a notation $i_1 - i_2 - ....i_r - i_1$. For example, Figure 7.5 shows two paths 2-4-5-2 (Figure 7.5 (a)) and 4-5-6-4 (Figure 7.5 (b))

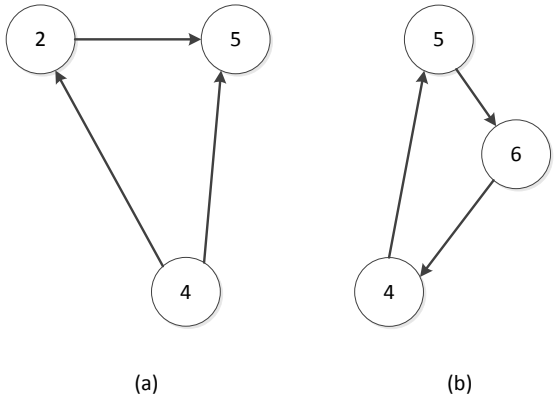


Figure 7.5: Examples of cycle

Directed cycle

A *directed cycle* is a directed path $i_1 - i_2 - ....i_{r-1} - i_r$ together with the edge (i_r, i_i). The graph shown in Figure 7.5(a) is a *cycle*, but not a *directed cycle*; the graph in Figure 7.5(b) is a *directed cycle*.

Connectivity

Two vertices *i* and *j* are said to be a *connected* if the graph contains at least one path from vertex *i* to vertex *j*. A graph is *connected* if every pair of its vertex is connected; otherwise, the graph is *disconnected*. The maximal connected subgraphs of a graph are referred to as its components. For instance, the graph shown in Figure 7.6(a) is a connected graph and the graph shown in Figure 7.6(b) is a disconnected graph. The later graph consists of the vertex sets {1,2,3,4} and {5,6}. A directed graph is called *weakly connected* if every two of its vertices are *connected*.

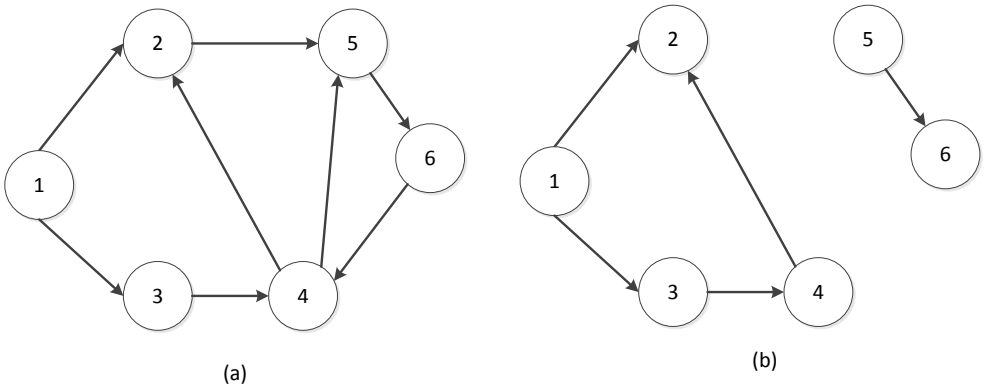
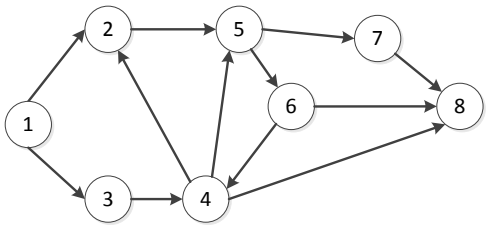


Figure 7.6: Disconnected and Connected graphs

Strong Connectivity

A *connected* graph is a *strongly connected* if it contains at least one *directed* path from every vertex to every other vertex. The strong components are the *maximal strongly connected* subgraphs. For the graph shown in Figure 7.1 reproduced here for convenience, {4, 6, 5, 2} is a set of strongly connected components.



Weak Connectivity

A directed graph is called *weakly connected* if every two of its vertices are *connected*. Thus, if there exists a uni directional arrow between two strongly connected components, then it can be said that there is a *weak connection* between these components. Note that both those components will be part of a single strongly connected component if the uni-directional arrow is replaced by an undirected edge (implying a bi-directional connectivity).

7.2 Proof of Privacy Protection in CLOPRO: A Graph Theoretic approach

In the CLOPRO framework, we define Privacy Protection as follows:

Privacy Protection Definition: The three components of privacy protection are Identity (*ID*), context (C_i) and original query (q_{orig_i}). For the purpose of proof of privacy protection we identify the four service parameters: Identity (ID), location at the time of request (**X**), start time of request (t_s) and the original query (q_{orig_i}). The context (C_i) comprises of location at time of request (**X**) and start time of request (t_s) . These parameters are called as Privacy Protection parameters. The privacy is said to be protected if it is not possible to establish a link between the Identity, the location, the time of query and the original query. If the link can be established, then it is said that Privacy is not protected.

The approach chosen here to demonstrate that the privacy is protected under a variety of attack modes is to create a directed graph and through it prove that the privacy is protected. Especially, our goal is to demonstrate that when the service request reference pair $< ID_m, ID_q >$ is encrypted, it is not possible to expose the identity of the user even if both the servers are compromised. Based on the graph theoretic framework, which has not been used in any of the past studies in this context, the various parameters in the system are represented by *vertices (V)* and the *edges (E)* between these parameters are represented as a one to one mapping via a database or a message string (communication from mobile device to either of the servers or

between the two servers (as defined by eq. 3.2, eq.3.3, eq.3.4). The directionality of an edge $e \in E$ depends on the feasibility of deciphering the '*head*' parameter in the event given that the '*tail*' parameter is known. In this setting, a *Strongly Connected Component (SCC)* of a graph represents a subset of parameters which are interlinked with each other in such a way that the knowledge of one can lead to the interpretation of all other parameters within that *SCC*. To this end, the privacy protection definition can be reformulated as:

- ♢ The privacy is said to be protected if none of the *SCC* of the Information Graph contain all the four privacy protection parameters, namely: $ID,\ \mathbf{X}_i,\ t_s,\ q_{orig_i}$.
- ♢ Otherwise, the privacy is not protected.

For the purpose of clarity, without any serious loss of generality henceforth $\mathbf{X}_i$ is referred to as $\mathbf{X}$, q_{orig_i} is referred to as q_{orig}, q_{code_i} is referred to as q_{code} and q_{gen_i} is referred to as q_{gen}. We use the well-known, standard Tarjan's algorithm [63] to identify the sets of Strongly Connected Components (*SCCs*) since it is an efficient algorithm with the worst case asymptotic complexity of $O(|V| + |E|)$.

Tarjan's Algorithm (Robert Tarjan) [63] is a well-established graph theory algorithm for identifying the *SCCs* of a graph. The algorithm requires a directed graph as an input. It produces a partition of the vertices of the graph into the *SCCs*. Each vertex of the graph appears in exactly a single set of the *SCCs*. Any vertex that is not on a directed cycle forms a *SCC* all by itself: for example, a vertex whose ***in-degree*** or ***out-degree*** is 0, or any vertex of an acyclic graph. The Tarjan's algorithm is based on *Depth First Search* (*DFS*). The algorithm runs the *DFS* search on two graphs, G and G^T: the *DAG* (*G*) in which the finish times for each vertex are recorded. The *DFS* algorithm is re-executed on (G^T) using the ordering obtained in the first phase. However, in the main loop of *DFS*, the vertices are considered in the order of decreasing finishing times, *f [u]* i.e. maximum finishing times first. Tarjan's algorithm is only a modified form of depth first search, hence it has an asymptotic complexity $O(|V| + |E|)$. Psuedocode for Tarjan's Algorithm is shown in Algorithm .

Algorithm 7.1 Tarjan's Algorithm (Source)

```
1: Input: Graph G=(V,E)
2: Output Set of strongly connected components (sets of vertices)
3:
4: index ← 0
5: S ← empty
6: for each v in V do
7:   if (v.index is undefined) then
8:     strongconnect (v)
9:   end if
10: end for
11:
12: function strongconnect(v)
13: //Set the depth index for v to the smallest unused index
14: v.index ← index
15: v.lowlink ← index
16: index← index + 1
17: S.push(v)
18: //Consider successors of v
19: for each (v, w) in E do do
20:   if (w.index is undefined) then
21:     //Successor w has not yet been visited; recurse on it
22:     strongconnect(w)
23:     v.lowlink ← min(v.lowlink, w.lowlink)
24:   else
25:     if (w is in S ) then
26:
27:       //Successor w is in stack S and hence in the current SCC
28:       v.lowlink ← min(v.lowlink, w.index)
29:     end if
30:   end if
31: end for
32: // If v is a root node, pop the stack and generate an SCC
33: if (v.lowlink = v.index) then
34:   //start a new strongly connected component
35:   repeat
36:     w ← S.pop()
37:     //add w to current strongly connected component
38:   until (w = v)
39:   // output the current strongly connected component
40: end if
```

We assess the privacy protection claim under various attack modes as follows (Figures show the types of attacks where the component in red is the component under attack.):

(i) no attack (on any server or no message string is sniffed). (Figure 7.7)

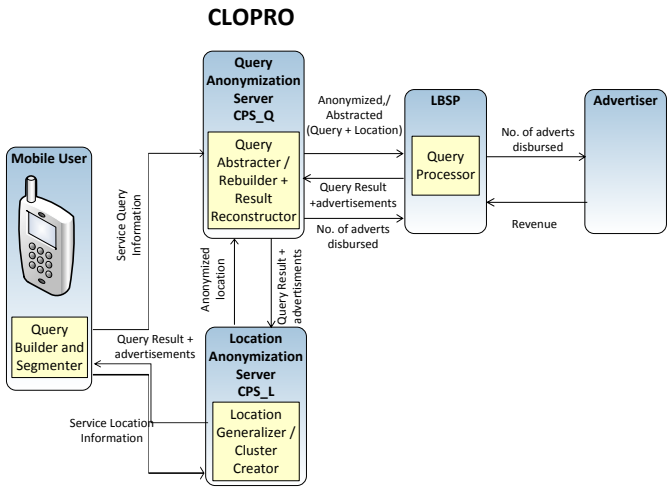
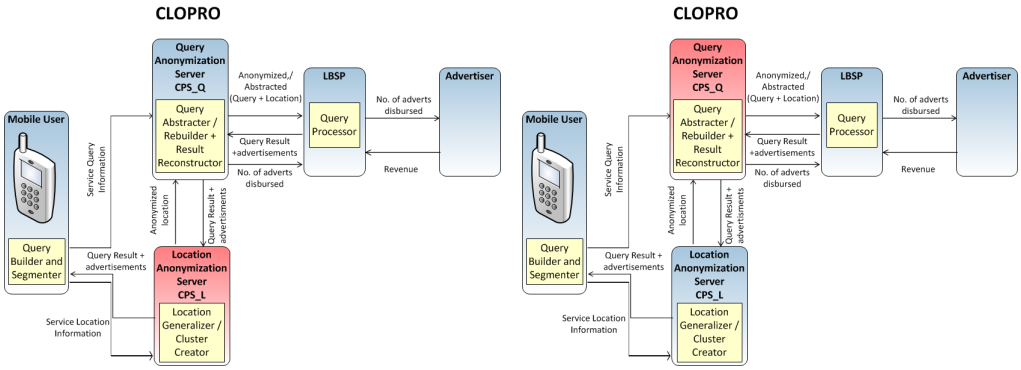


Figure 7.7: No Attack

(ii) attack on any one server (CPS_Q or CPS_L) as seen in Figure 7.8.



(a) Attack on CPS_L **(b)** Attack on CPS_Q

Figure 7.8: Attack on any one server

(iii) attack on both the servers simultaneously(CPS_Q and CPS_L) as seen in Figure 7.9.

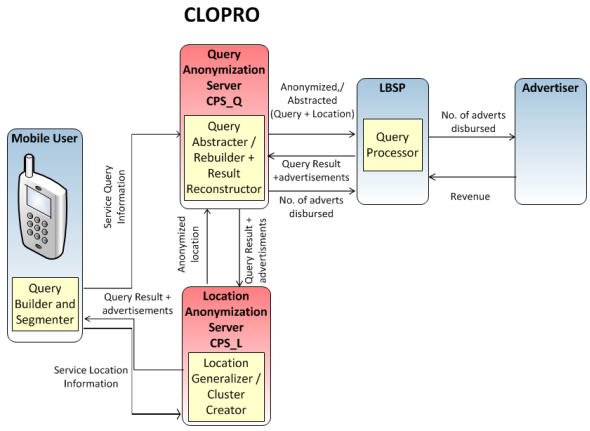


Figure 7.9: Attack on both servers

(iv) attack on a single message string to each of the servers (CPS_Q or CPS_L) or between the two servers as seen in Figure 7.10.

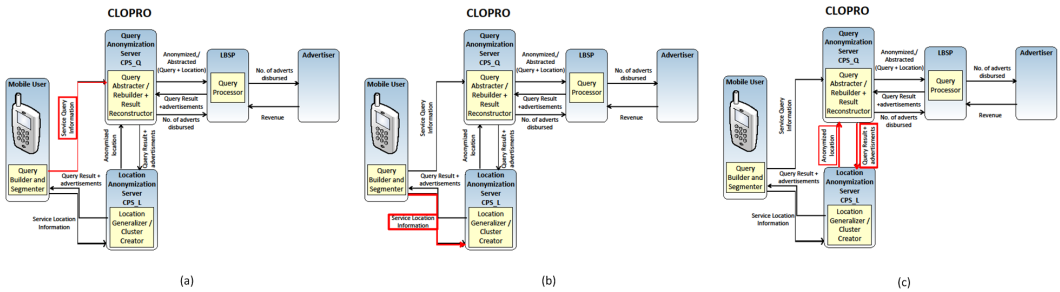


Figure 7.10: Attack on Message strings

(v) attack on the message strings (all cases of type (iv)) along with attack on any one server (CPS_Q or CPS_L) as seen in Figure .

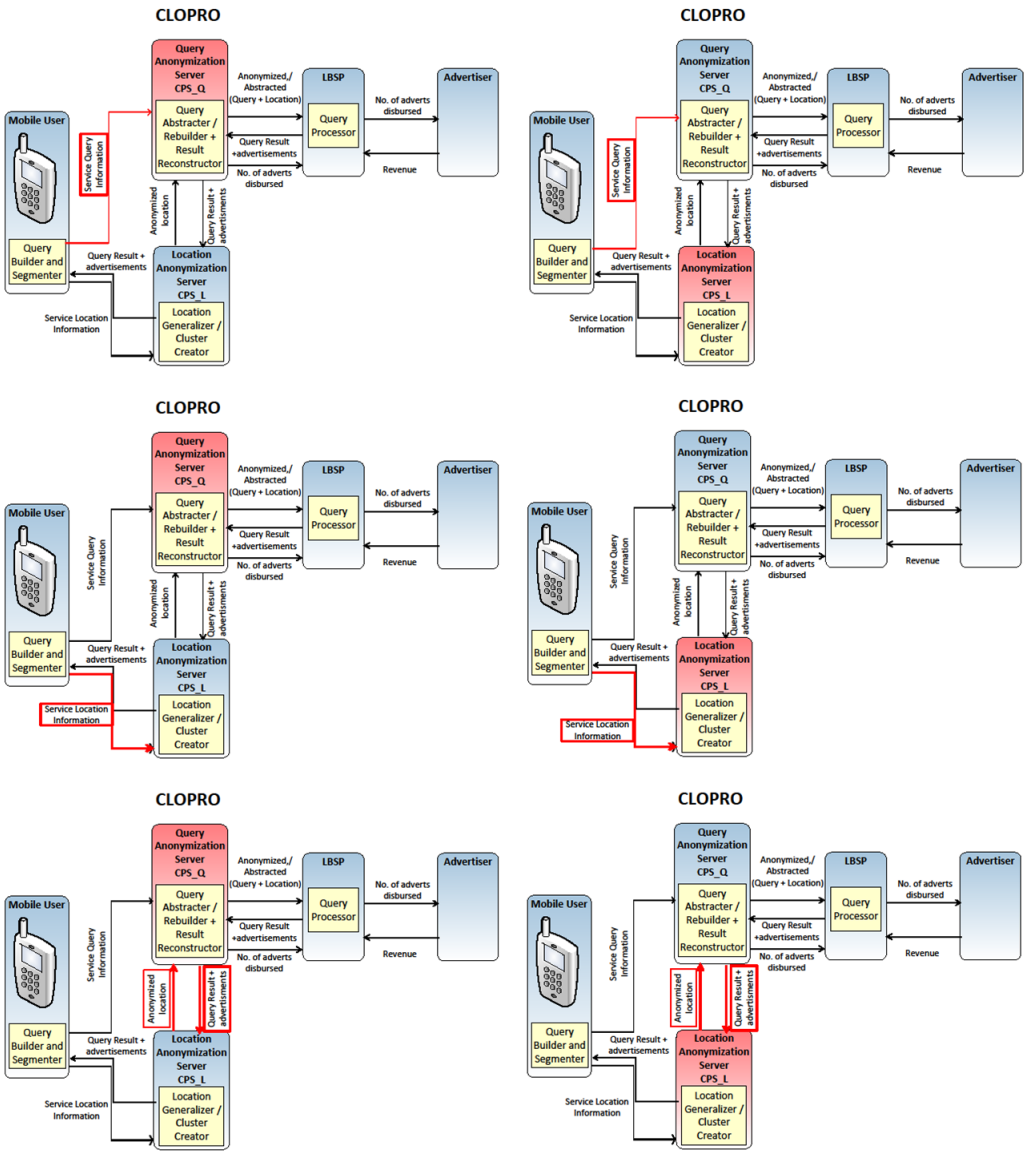


Figure 7.11: Attack on a message string and any one server

All odd numbered cases (1, 3, 5..) are situations with ID_m and ID_q are unencrypted and even numbered cases (2, 4, 6..) are the situations with ID_m and ID_q encrypted. The results depicted and discussed below show that the privacy is not protected under three conditions when the ID_m and ID_q are unencrypted:

1. When both the servers are compromised simultaneously
2. When the CPS_Q server is attacked along with the message string to the CPS_L server and

3. When the CPS_L server is attacked along with the message string to the CPS_Q server

Note that the CLOPRO framework encrypts this unique request reference pair ID_m and ID_q, and in those cases, the ID is protected under all conditions. For each of the cases, various vertices (parameters that define the identity, the query, the location and the time) in the system are as follows:

Symbol	**Description**
ID	User's identity
ID_m	Mobile device generated unique user id for every service request
ID_q	Mobile device generated unique query id for every service request
X	Location coordinates of the device at time of issue of query
$\mathbf{X}_{clust}$	Location coordinates of the cluster of k users within a certain area having the same query.
t_s	Time of issue of user query
t_f	Desired response time for the query
q_{orig}	Original query of the user
q_{gen}	Generic query for the original query of the user
q_{code}	Code for the generic query associated with the original query of the user
ID_{clust}	A unique id issued by the CPS_Q server for the cluster of users before the query is forwarded to the LBSP
R	The response to the query given by the LBSP

The color coding to represent the graphs is:

Green: Not possible to decipher the parameter.

Yellow: Possible to decipher the parameter under certain conditions.

Red: Parameter will be deciphered since it is part of *SCC* with parameters sent to the service provider.

In the explanation of the cases to follow, the approach used is as follows. The input graph will show the connections between the parameters. The arrowhead will indicate that the parameter can be deciphered if the parameter at the tail end is known. A bidirectional arrow between two parameters indicates that knowledge of one parameter can be obtained from the knowledge of the other and vice versa. The solid arrows indicate that the relationship between these parameters is known in case when none of the servers are attacked or none of the message strings are sniffed (No attack) i.e., this information is decipherable to the adversary based on

the knowledge of the message string going to the LBSP and the knowledge of the parameter at the tail-end of the arrow. The dotted arrows will indicate that those parameters will be exposed under special conditions of attack, for eg. attack on a particular server or attack on the message strings. The input file to the Tarjan algorithm will depict these relations in the following form: If there exist an unidirectional arrow, say between $ID_m \rightarrow ID$, then the input file will contain a row $ID_m\ ID$. However, if a bidirectional arrow exists between these two parameters ($ID_m \leftrightarrow ID$) then two entries on separate rows will be made $ID_m\ ID$ and $ID\ ID_m$. Before we discuss the cases, Table 7.1 presents a short summary of all the cases with < ID_m, ID_q> encrypted and subsequently discussed in details in the following sections.

Table 7.1: Summary of various cases under consideration with $< ID_m, ID_q >$ encrypted

Case No.	Condition	Protected	Result
2	"No Attack", <ID_m, ID_q> encrypted	ID,**X**, t_s, q_{orig}	Safe
4	"CPS_Q Attack", <ID_m, ID_q> encrypted	ID,**X**, t_s	Safe
6	"CPS_L Attack", <ID_m, ID_q> encrypted	ID,q_{orig}	Safe
8	"CPS_Q and CPS_L Attack", <ID_m, ID_q> encrypted	ID	Safe
10	"Message string to CPS_Q server", <ID_m, ID_q> encrypted	ID,**X**, t_s	Safe
12	"Message string to CPS_L server", <ID_m, ID_q> encrypted	ID,q_{orig}	Safe
14	"Message string between CPS_L and CPS_Q server", <ID_m, ID_q> encrypted	ID,**X**, q_{orig}	Safe
16	"Message string to CPS_Q server + CPS_Q server", <ID_m, ID_q> encrypted	ID,**X**, t_s	Safe
18	"Message string to CPS_Q server + CPS_L server", <ID_m, ID_q> encrypted	ID	Safe
20	"Message string to CPS_L server + CPS_Q server", <ID_m, ID_q> encrypted	ID	Safe
22	"Message string to CPS_L server + CPS_L server", <ID_m, ID_q> encrypted	ID,q_{orig}	Safe
24	"Message string between CPS_L and CPS_Q server + CPS_Q server", <ID_m, ID_q> encrypted	ID,**X**	Safe
26	"Message string between CPS_L and CPS_Q server + CPS_L server", <ID_m, ID_q> encrypted	ID,q_{orig}	Safe

The summary table 7.1 shows that under no conditions, all the privacy protection variables are revealed. The most interesting cases (numbered 8, 18, 20) are the situations that are similar to the mobile device being attacked or that the adversary has access to all the data as in case of a global observer. In all these cases, when both servers are attacked simultaneously (case 8), or one server along with the message string going to the server other than the one compromised is attacked (cases 18 and 20) the adversary effectively has access to all the information. In those cases, it would seem that the servers are colluding or the mobile device itself is compromised.

Even in those cases, it can be seen that the ID of the user is protected making this a safe way to obtain the location based services while protecting the privacy of the user. Note that we are reducing the overhead and improving the efficiency by simply encrypting just the unique request reference pair <ID_m, ID_q> and not the whole message string as in case of [11].

For the sake of completeness, we have also examined all the cases where the <ID_m, ID_q> is not encrypted. The outcomes of those cases are summarized in Table 7.2. As the table shows, excepts in cases 7, 17 and 19, there is at least one privacy parameter protected, implying that it is not possible to establish a link between all the four privacy parameters. These conditions are the ones where both the servers are compromised simultaneously or one server along with the message string to the other non-compromised server is attacked by the adversary. These situations are similar to the mobile device itself being attacked and thus cannot be protected from the adversary. So it can be seen that even if the unique request reference pair is not encrypted, in most of the cases the privacy of the user is protected.

Table 7.2: Summary of various cases under consideration with $< ID_m, ID_q >$ unencrypted

Case No.	Condition	Protected	Result
1	"No Attack", <ID_m, ID_q> unencrypted	**X**, t_s, q_{orig}	Safe
3	"CPS_Q Attack", <ID_m, ID_q> unencrypted	**X**, t_s	Safe
5	"CPS_L Attack", <ID_m, ID_q> unencrypted	q_{orig}	Safe
7	"CPS_Q and CPS_L Attack", <ID_m, ID_q> unencrypted	-	Unsafe
9	"Message string to CPS_Q server", <ID_m, ID_q> unencrypted	**X**, t_s	Safe
11	"Message string to CPS_L server", <ID_m, ID_q> unencrypted	q_{orig}	Safe
13	"Message string between CPS_L and CPS_Q server", <ID_m, ID_q> unencrypted	**X**, q_{orig}	Safe
15	"Message string to CPS_Q server + CPS_Q server", <ID_m, ID_q> unencrypted	**X**, t_s	Safe
17	"Message string to CPS_Q server + CPS_L server", <ID_m, ID_q> unencrypted	-	Unsafe
19	"Message string to CPS_L server + CPS_Q server", <ID_m, ID_q> unencrypted	-	Unsafe
21	"Message string to CPS_L server + CPS_L server", <ID_m, ID_q> unencrypted	q_{orig}	Safe
23	"Message string between CPS_L and CPS_Q server + CPS_Q server", <ID_m, ID_q> unencrypted	**X**	Safe
25	"Message string between CPS_L and CPS_Q server + CPS_L server", <ID_m, ID_q> unencrypted	q_{orig}	Safe

7.2.1 Set 1: No Attack on any server or any message string

Case 1: None of the servers or message strings are attacked (with ID_m and ID_q unencrypted):

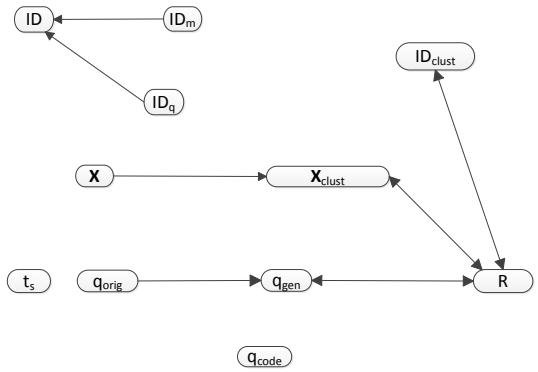


Figure 7.12: Input graph Under "No Attack" condition

Input Graph

The input graph of Figure 7.12 shows the connections between various parameters (vertices). As mentioned earlier, if an edge exists between two parameters, it indicates that the parameter at the arrowhead can be discovered if the parameter at tail end is known. The graph above shows that if the attacker is able to obtain the ID_m or ID_q (and has knowledge of the methodology used to generate them) it is possible to decipher the ID of the user. But identification of ID_mor ID_q even if ID is disclosed, is not possible since each service request has generated a unique pair of ID_q and ID_m comprising of a random component. Hence, there is only a unidirectional arrow between ID_m, ID_q and ID . Even though we have indicated the unidirectional arrow $ID_q \rightarrow ID$ or $ID_m \rightarrow ID$ note that, it is not possible to decipher it if the adversary gets a single snapshot of these values. The probability of discovering the relation is only when the adversary has multiple snapshots and is able to identify the pattern of the user's ID. Similarly, when the location of the mobile device$\mathbf{X}$ is known for all the users it is possible to find the ($\mathbf{X}_{clust}$) but the reverse is not possible. However, the edges between ID_{clust}, ($\mathbf{X}_{clust}$), q_{gen} and R are bidirectional as it is possible to find R from each one of these parameters and vice versa. The input graph connections are as follows:

Connections in Input Graph:

Parameter	Connectivity	Parameter
$\mathbf{X}$	$\rightarrow$	$\mathbf{X}_{clust}$
q_{orig}	$\rightarrow$	q_{gen}
$\mathbf{X}_{clust}$	$\leftrightarrow$	R
q_{gen}	$\leftrightarrow$	R
ID_{clust}	$\leftrightarrow$	R
ID_m	$\rightarrow$	ID
ID_q	$\rightarrow$	ID

The input file *NoAttack.in* is as shown in Figure 7.13.

NoAttack.in

1	ID_m	ID
2	ID_q	ID
3	$\mathbf{X}$	$\mathbf{X}_{clust}$
4	q_{orig}	q_{gen}
5	q_{gen}	R
6	R	q_{gen}
7	$\mathbf{X}_{clust}$	R
8	R	$\mathbf{X}_{clust}$
9	ID_{clust}	R
10	R	ID_{clust}

Figure 7.13: Example of Input File under "No Attack" condition

Tarjan's algorithm is applied to this input file.

Output:

The output of Tarjan's algorithm shows that there are six sets of *SCCs* under the condition that none of the message strings or the servers are attacked:

Set 0: [ID]

Set 1: [ID_m]

Set 2: [ID_q]

Set 3: [ID_{clust}, q_{gen}, R, $\mathbf{X}_{clust}$]

Set 4: [$\mathbf{X}$]

Set 5: [q_{orig}]

Figure 7.14 shows the output graph indicating these sets.

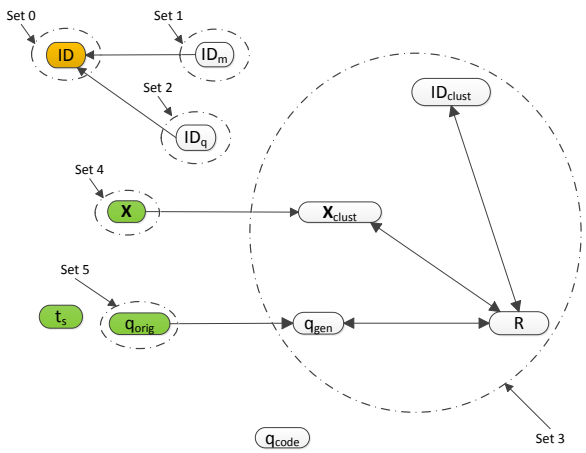


Figure 7.14: Output graph under "No Attack" condition

Each of the ID, ID_m, ID_q, $\mathbf{X}$ and q_{orig} parameters are not identifiable from any other vertices since each is a separate *SCC*. Thus, based on the knowledge available to the attacker (indicated through the edges of the graph) it is not possible to decipher any of the above privacy protection parameters by the attacker. There is a connection between $\mathbf{X}$ (Set 4) and Set 3. However, since the direction of arrow is from $\mathbf{X}$ to Set 3, it is not possible to identify the $\mathbf{X}$ if all parameters in Set 3 are known. Even if the attacker has the knowledge of the fact that this query was asked from this location $\mathbf{X}$, it is not possible to associate it to a particular user (ID) and the time the service request has been initiated (t_s). The parameters ID_{clust}, ($\mathbf{X}_{clust}$), q_{gen} and R form a set of *SCC*s. If the attacker has the knowledge of any one of these parameters, it is possible to decipher details regarding any of the other parameters within *SCC*. However, this does not reveal link between any of the privacy protection parameters such as ID, $\mathbf{X}$, t_s and q_{orig}. The connection between the *SCC* Set 1 and *SCC* Set 0 $ID : ID_m$ and the connection between *SCC* Set 2 and *SCC* Set 0$ID : ID_q$ indicate that it is possible for the attacker to identify the ID if the mechanism for generation of ID_m, ID_q is known to the attacker. The parameters $\mathbf{X}$, t_s and q_{orig} shown in the green color are not a part of any *SCC* set and thus it can be said that they are not discoverable by the attacker. **The basic premise of privacy protection is that the link between ID, $\mathbf{X}$, t_s and q_{orig} cannot be established, the privacy is stated to be protected.**

In the cases to follow, only the output graphs will be shown. The input graph and the corresponding input file will be assumed to have been formed in a manner similar to that shown in Case 1.

Case 2: None of the servers or message strings are attacked (with ID_m and ID_q encrypted):

Input Graph:

Connections in Input Graph:

Parameter	Connectivity	Parameter
$\mathbf{X}$	$\rightarrow$	$\mathbf{X}_{clust}$
q_{orig}	$\rightarrow$	q_{gen}
$\mathbf{X}_{clust}$	$\leftrightarrow$	R
q_{gen}	$\leftrightarrow$	R
ID_{clust}	$\leftrightarrow$	R

Note that in this case we have ID_m and ID_q are encrypted and so there is no connections between $ID_m$and ID and ID_q and ID as in Case 1. An input file is created based on these connections. Tarjan's algorithm is applied to this input file.

Output

The output of Tarjan's algorithm shows that there are three sets of *SCCs* and Figure 7.15 shows the output graph indicating the *SCC*s, under the condition that none of the message strings or the servers are attacked and the ID_m and ID_q are encrypted:

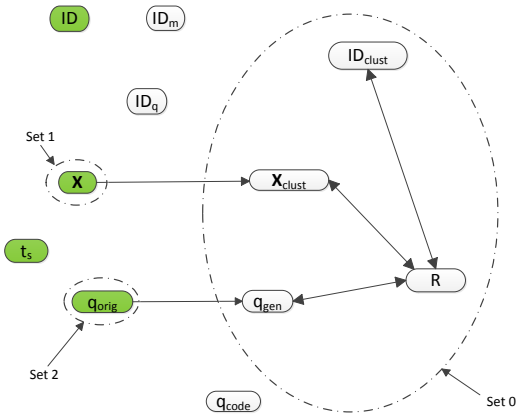


Figure 7.15: Output graph under "No Attack" condition with ID_m and ID_q encrypted

The Set 0 shows that the parameters ID_{clust}, ($\mathbf{X}_{clust}$), q_{gen} and R belong to one set of *SCC*s. If the attacker can get information regarding any one of these parameters it is possible to decipher details regarding any of the other parameters. However, this does not reveal any of the

privacy parameters. Set 1 consists of only the location coordinates, **X** and Set 2 has only q_{orig}. There is a connection between **X** (Set 1) and Set 0. Also, there is a connection between q_{orig} (Set 2) and Set 0. However, since the direction of arrow is from **X** to Set 0 and q_{orig} to Set 0, it is not possible to identify the **X** and q_{orig} if all parameters in Set 0 are known. Even if the attacker has the knowledge of the fact that this query (q_{orig}) was asked from this location **X**, it is not possible to associate the same to a particular user (ID) and the time of initiation of this service request (t_s) as they are not connected with Set 0. All the relevant parameters shown in green color: ID, **X**, t_s and q_{orig} are not discoverable by the attacker under this condition. **Thus, based on the definition, privacy is said to be protected**.

In the discussions that follow, Case 1 is treated as a base case for all the odd numbered cases and Case 2 is treated as a base case for all the even numbered cases. Since the size of the files increase due to additional connections with the attacks on the servers or the attack on the message strings, only connections that are added due to attacks are shown.

7.2.2 Set 2: Attack on any one server (CPS_Q or CPS_L)

Case 3: Server CPS_Q is attacked (with ID_m and ID_q unencrypted):

Input Graph:

In this case, we consider a scenario when the CPS_Q server is attacked. This implies that an adversary has a control over the Qcode ($ID, q_{orig}, q_{gen}, q_{code}$), Uid ($ID_{mk}, ID_{clust}, ID_{q_k}$) and User_query ($ID_m, ID_q, q_{code}, q_{orig}, t_f$) databases. The other databases stored on the CPS_Q server are not considered here for the following reasons: LBSP_QueryFormat, LBSP_Services, LBSP_Info consist of information related to the LBSP, User_Interests and User_Apps do not have any other information and connectivity.

The additional connections in the Input Graph are :

Parameter	Connectivity	Parameter
ID	$\leftrightarrow$	q_{orig}
ID	$\leftrightarrow$	q_{gen}
ID	$\leftrightarrow$	q_{code}
ID_m	$\leftrightarrow$	ID_{clust}
ID_m	$\leftrightarrow$	ID_q
ID_{clust}	$\leftrightarrow$	ID_q
ID_m	$\leftrightarrow$	q_{code}
ID_m	$\leftrightarrow$	q_{orig}
ID_m	$\leftrightarrow$	t_f
ID_q	$\leftrightarrow$	q_{code}
ID_q	$\leftrightarrow$	q_{orig}
ID_q	$\leftrightarrow$	t_f
q_{code}	$\leftrightarrow$	q_{orig}
q_{code}	$\leftrightarrow$	t_f
q_{orig}	$\leftrightarrow$	t_f

An input file is created based on these connections. Tarjan's algorithm is applied to this input file.

Output

The output of Tarjan's algorithm shows that there are three sets of *SCCs* and Figure 7.16 shows the output graph indicating the *SCC*s under the condition that only the CPS_Q server is attacked.

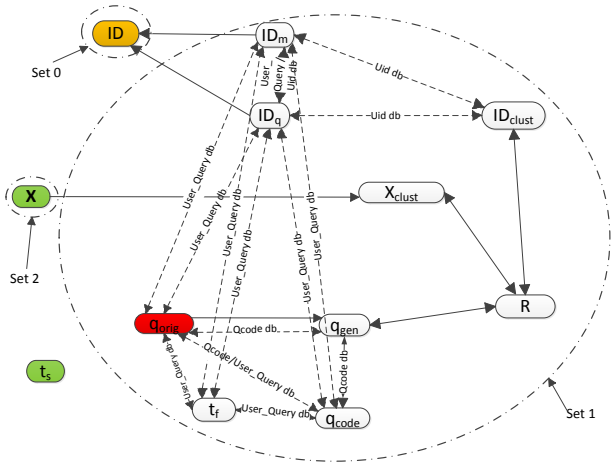


Figure 7.16: Output graph under "CPS_Q Attack" condition

SCC Set 0 consists of only ID, and Set 2 consists of only **X**. Set 1 contains ID_{clust}, X_{clust}, R,t_f, q_{orig}, q_{gen}, q_{code}, ID_q, ID_m. The unidirectional arrows from ID_m and ID_q to ID indicate that ID can be deciphered from the knowledge of any parameter in Set 1. Hence it is shown yellow in color. The arrow between **X** (Set 2) and Set 1 is directed from **X** to Set 1. Thus, knowledge of any element in Set 1 does not reveal the coordinates **X** of the user and hence undecipherable so shown in green. The parameter t_s is not a part of any *SCC* and hence undecipherable, so shown in green. However, the parameter q_{orig} can be obtained from the knowledge of any parameter in Set 1. Since the ID_{clust} and q_{gen} are known to the service provider, it is possible to decipher q_{orig}. Hence, it is shown in red. Although q_{orig} can be deciphered by the attacker, it is not possible to obtain the link between the four privacy protecting parameters since it is not possible to decipher **X** and t_s. **Thus, it can be safely assumed that privacy is protected even if the CPS_Q server is attacked.**

Case 4: Server CPS_Q is attacked (with ID_m and ID_q encrypted):

Input Graph:

As in the earlier case 2, the ID_m and the ID_q are in encrypted form even when CPS_Q is attacked. The ID_m and the ID_q are encrypted on the mobile device. Hence the only difference between case 3 and case 4 is that the edges between the $ID_m \rightarrow ID$ and $ID_q \rightarrow ID$ do not exist. An input file is created based on these connections. Tarjan's algorithm is applied to this input file.

Output:

The output shows that there are just two sets of *SCC*s as shown in Figure 7.17:

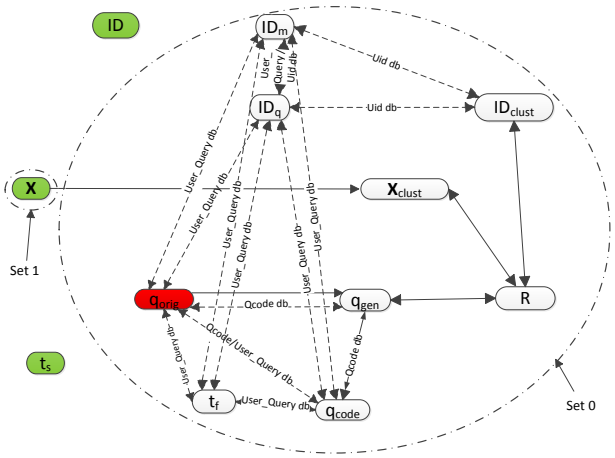


Figure 7.17: Output graph under "CPS_Q Attack" condition with ID_m and ID_q encrypted

The direction of the arrow between **X** and Set 0 implies that the knowledge of the parameters in Set 0 will not reveal **X** (and hence is shown in green color), the coordinates of the user. Since the parameters ID and t_s are not part of any *SCC* set, they are not decipherable and hence shown in green color. As discussed in Case 3, the q_{orig} is decipherable by the attacker and hence is indicated in red. Although q_{orig} can be deciphered by the attacker, it is not possible to obtain the link between the four privacy protecting parameters since it is not possible to decipher ID, **X** and t_s. Thus, it can be safely assumed that the privacy is protected even if the CPS_Q server is attacked when the service request reference pair $< ID_m,\ ID_q >$ is encrypted.

Case 5: Server CPS_L is attacked (with ID_m and ID_q unencrypted):

Input Graph:

In this case, we consider a scenario when the CPS_L server is attacked. This implies that an adversary has a control over the Loc_info database ($ID_q, q_{code}, k, \mathbf{X}, \mathbf{X}_{clust}, t_s, t_f$). If the adversary can successfully get access to CPS_L server, the adversary has access to ($ID_q, q_{code}, k, \mathbf{X}, \mathbf{X}_{clust}, t_s, t_f$) parameters. All of these parameter connections are shown using bidirectional dotted arrows between each of these parameters since each of the parameters in the database are obtained when one of them is known.

The additional connections in the Input Graph of Case 5 is

Parameter	Connectivity	Parameter
ID_q	$\leftrightarrow$	q_{code}
ID_q	$\leftrightarrow$	k
ID_q	$\leftrightarrow$	$\mathbf{X}$
ID_q	$\leftrightarrow$	$\mathbf{X}_{clust}$
ID_q	$\leftrightarrow$	t_s
ID_q	$\leftrightarrow$	t_f
q_{code}	$\leftrightarrow$	k
q_{code}	$\leftrightarrow$	$\mathbf{X}$
q_{code}	$\leftrightarrow$	$\mathbf{X}_{clust}$
q_{code}	$\leftrightarrow$	t_s
q_{code}	$\leftrightarrow$	t_f
k	$\leftrightarrow$	$\mathbf{X}$
k	$\leftrightarrow$	$\mathbf{X}_{clust}$
k	$\leftrightarrow$	t_s
k	$\leftrightarrow$	t_f
$\mathbf{X}$	$\leftrightarrow$	$\mathbf{X}_{clust}$
$\mathbf{X}$	$\leftrightarrow$	t_s
$\mathbf{X}$	$\leftrightarrow$	t_f
$\mathbf{X}_{clust}$	$\leftrightarrow$	t_s
$\mathbf{X}_{clust}$	$\leftrightarrow$	t_f
t_s	$\leftrightarrow$	t_f

An input file is created based on these connections. Tarjan's algorithm is applied to this input file.

Output:

The output shows that there are four sets of *SCC*s as shown in Figure 7.18:

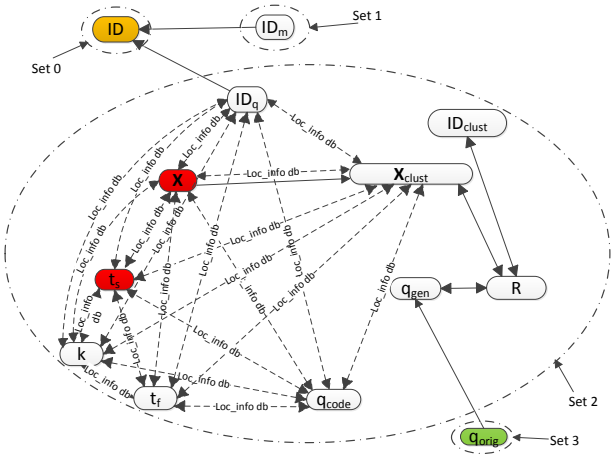


Figure 7.18: Output graph under "CPS_L Attack" condition

As can be seen Set 0 has only ID, Set 1 has only ID_m and Set 3 only has q_{orig}. Since $\mathbf{X}$ and t_s are decipherable by adversary with the knowledge of ($\mathbf{X}_{clust}$), a parameter known to the LBSP, they are indicated with the red color. The ID is shown yellow since the knowledge of ID_q from Set 1 can make ID decipherable. However, q_{orig} is shown in green since the direction of the arrow between Set 3 and Set 2 is from q_{orig} to Set 2. Since q_{orig} parameter is not decipherable with attack on CPS_L even without the service request reference pair $< ID_m, ID_q >$ are unencrypted, it can be said that **privacy is protected in the case** when CPS_L server is attacked.

Case 6: Server CPS_L is attacked (with ID_m and ID_q encrypted):

Input Graph:

In this case, we consider a scenario when the CPS_L server is attacked, but the unique service request reference pair is encrypted. When an adversary successfully gets access to the CPS_L server, the adversary has access to ($ID_q, q_{code}, k, \mathbf{X}, \mathbf{X}_{clust}, t_s, t_f$) parameters. The only difference between Case 5 and Case 6 is that the connections $ID_m \rightarrow ID$ and $ID_q \rightarrow ID$ are removed. The remaining input file will consist of entries similar to those of the input file for Case 5.

Output:

The output shows that there are two sets of *SCC*s as shown in Figure 7.19:

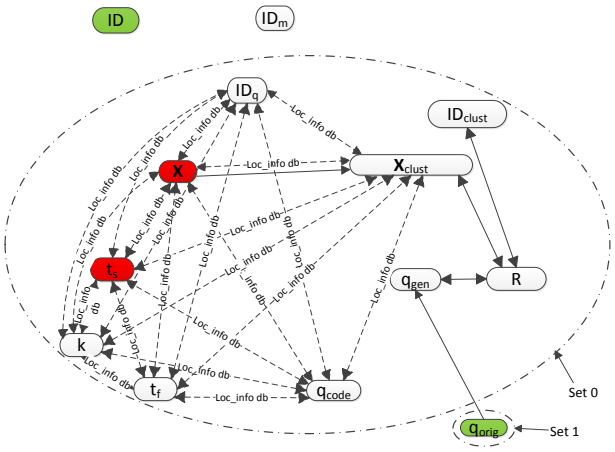


Figure 7.19: Output graph under "CPS_L Attack" condition with ID_m and ID_q encrypted

Set 1 consists of q_{orig}. Although there is a unidirectional arrow connecting q_{orig} and Set 0, the direction renders q_{orig} as undecipherable so it is indicated in green color. ID is not a part of any *SCC* so it is undecipherable and is shown in green color. Since **X** and t_s are a part of Set 0 *SCC,* they are decipherable by the attacker and hence are shown in red color. The knowledge of the location (**X**) and time of query (t_s) does not expose the user (ID and the original query q_{orig} are not exposed). Since two of the four privacy protection parameters, ID and q_{orig} are not disclosed, it can be said that **the privacy is protected under the condition that server CPS_L server is attacked and the service request reference pair $< ID_m, ID_q >$ is encrypted (as per the scheme of the proposed framework)**.

7.2.3 Set 3: Both Servers are attacked simultaneously (CPS_Q and CPS_L)

Case 7: Both the servers CPS_Q and CPS_L are attacked simultaneously by an adversary (with ID_m and ID_q unencrypted):

Input Graph:

If both the servers CPS_Q and CPS_L are attacked, the parameters from all the databases in both the servers is exposed to the adversary. The input graphs of Case 3 and Case 5 are superimposed to create an input graph for the case when both the servers are attacked. All

of these parameter connections are shown using bidirectional dotted arrows between each of these parameters. An input file corresponding to the connections in the graph is created. The input file comprises of connections specified in Case 1, Case 3 and Case 5. Tarjan's algorithm is applied to this input file.

Output:

The output shows that there are two sets of *SCC*s as shown in Figure 7.20:

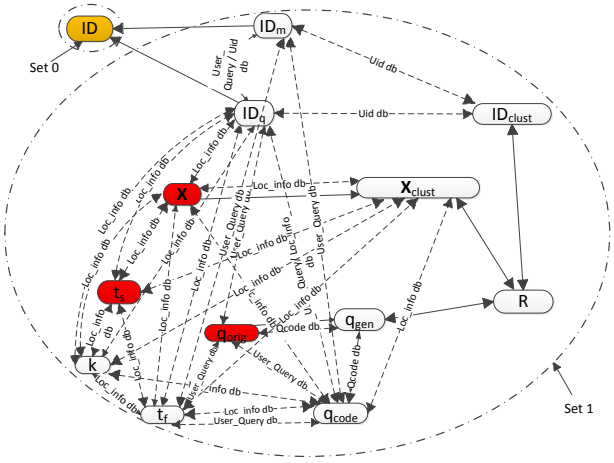


Figure 7.20: Output graph under "CPS_Q and CPS_L Attack" condition

Set 0 contains only theID and Set 1 *SCC* contains all other elements. Since there is a unidirectional arrow fromID_m / ID_q to ID of Set 0 it is possible to decipher the ID through the connection between these parameters. This is expected, when both the servers are attacked, there is maximum possibility of threat to privacy and all the parameters will be disclosed. Thus, to ensure the privacy even under the condition that both servers are attacked, there needs to be some additional precaution which is taken care in the process of the CLOPRO framework and is discussed in the next case. **Thus, this condition will be a threat to the privacy protection of the user.**

Case 8: Both the servers CPS_Q and CPS_L are attacked simultaneously by an adversary(with ID_m and ID_q encrypted):

Input Graph:

If both the servers CPS_Q and CPS_L are attacked with the unique service request reference pair is encrypted, the parameters stored in all the databases on both these servers is exposed to the adversary,. The input graphs of Case 4 and Case 6 are superimposed to create an input

graph for the case when both the servers are attacked. All of these parameter connections are shown using bidirectional dotted arrows between each of these parameters. An input file corresponding to the connections in the graph is created. The input file comprises of connections specified in Case 2, Case 4 and Case 6. Tarjan's algorithm is applied to this input file.

Output:

The output shows that there are only one set of *SCC* as shown in Figure 7.21:

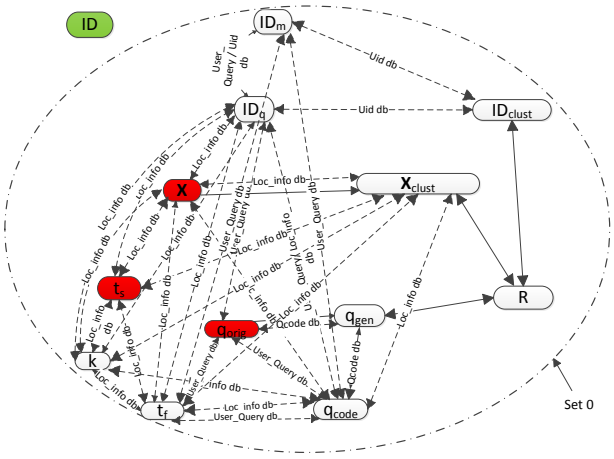


Figure 7.21: Output graph under "CPS_Q and CPS_L Attack" condition with ID_m and ID_q encrypted

The output graph shows that only one parameter ID is not included in the single *SCC*. Since there is no connection of this component with the *SCC*, it is not decipherable and hence it is marked with green color. As seen from the graph, the privacy parameters **X** , t_s and q_{orig} are part of the *SCC* and so are decipherable by the adversary. **Thus, it can be seen that even if both the servers are attacked, when CLOPRO framework is used, it is not possible to obtain the identity of the user (ID), which is the most important aspect in privacy protection of the user.**

The graphs will not differ from the above cases even when the message strings are also attacked along with the servers since the information in the message strings is stored in the respective databases on the two servers.

7.2.4 Set 4: Message strings to any one of the servers, CPS_Q or CPS_L, or between the two servers is attacked

Case 9: Message string to server CPS_Q is attacked by an adversary (with ID_m and ID_q unencrypted):

Input Graph:

Next case considered is when the attacker sniffs the message string to server CPS_Q (q_{CPS_Q}, as defined in eq. 3.2). Apart from the basic edges of the graph from Case 1, bi-directional arrows are included to show the input connections between the parameters ID_m, ID_q, q_{orig}, q_{code} and t_f. which are parameters of the message string to the CPS_Q server. Thus, the connections, in addition to those in Case 1 are as shown

Parameter	Connectivity	Parameter
ID_q	$\leftrightarrow$	ID_m
ID_q	$\leftrightarrow$	q_{orig}
ID_q	$\leftrightarrow$	q_{code}
ID_q	$\leftrightarrow$	t_f
ID_m	$\leftrightarrow$	q_{orig}
ID_m	$\leftrightarrow$	q_{code}
ID_m	$\leftrightarrow$	t_f
q_{orig}	$\leftrightarrow$	q_{code}
q_{orig}	$\leftrightarrow$	t_f
q_{code}	$\leftrightarrow$	t_f

An input file corresponding to the connections in the graph is created. Tarjan's algorithm is applied to this input files.

Output:

The output shows that there are four sets of *SCC* as shown in Figure 7.22

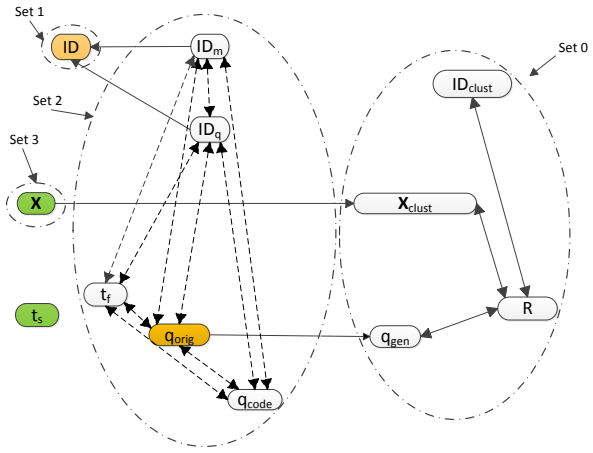


Figure 7.22: Output graph under "Attack on message string to CPS_Q" condition

Set 1 consists of ID and Set 3 consists of X. The other two sets: Set 0 contains parameters known to the LBSP and Set 2 contains ID_m, ID_q, q_{orig}, q_{code} and t_f . Since it is possible to decipher q_{orig}, which is part of Set 2, from the knowledge of ID_m, ID_q making it decipherable under the condition that the ID can be deciphered through connection with ID_m or ID_q and hence marked with orange color. Since it is also possible to decipher ID from knowledge of ID_q/ID_m, we indicate those parameters on the graph in orange color too. Since t_s is not a part of any set of *SCC*, it is not decipherable and hence shown in green color. Also, since the arrow head rests on Set 0, it is not possible to decipher **X** and so is shown green in color. **As seen in the graph, it is not possible to envisage connections against all four privacy parameter (X and t_s are not discoverable), it can be safely said that users privacy is preserved.**

Case 10: Message string to server CPS_Q is attacked by an adversary (with ID_m and ID_q encrypted):

Input Graph:

Next case considered is when the attacker sniffs the message string to server CPS_Q (q_{CPS_Q}) with ID_m and ID_q encrypted. Apart from the basic edges of the graph from Case 2, bi-directional arrows are included to show the input connections between the parameters ID_m, ID_q, q_{orig}, q_{code} and t_f which are parameters of the message string to the CPS_Q server. An in-

put file corresponding to the connections in the graph is created. Tarjan's algorithm is applied to this input files.

Output:

The output shows that there are three sets of *SCC* as shown in Figure 7.23

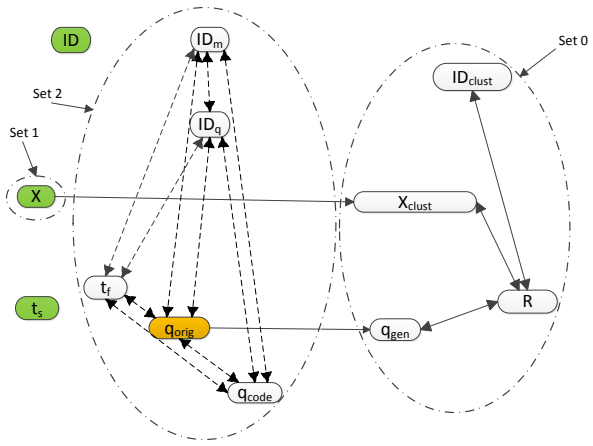


Figure 7.23: Output graph under "Attack on message string to CPS_Q" condition with ID_m and ID_q encrypted

As can be seen from the graph, theID, **X** and t_s are not decipherable from any *SCC* and hence depicted in green color. It is possible to decipher q_{orig} parameter as it is a part of the *SCC* Set 2 that is part of the message string sent to CPS_Q server. **Since three out of four privacy parameters are not decipherable (**ID**,** **X** **and**t_s **), it can be said that privacy is protected if the message string to CPS_Q server is sniffed by the adversary and** ID_m **and** ID_q **are encrypted.**

Case 11: Message string to server CPS_L is attacked by an adversary (with ID_m **and** ID_q **unencrypted):**

Input Graph:

Next case considered is when the attacker sniffs the message string to server CPS_L (q_{CPS_L} as defined in eq. 3.3). Apart from the basic edges of the graph from Case 1, bi-directional arrows are included to show the input connections between the parameters ID_q, q_{code}, k, **X** , t_s and t_f which are the parameters of the message string to the CPS_L server. Thus, the connections, in addition to those in Case 1 are as shown

Parameter	Connectivity	Parameter
ID_q	$\leftrightarrow$	q_{code}
ID_q	$\leftrightarrow$	k
ID_q	$\leftrightarrow$	$\mathbf{X}$
ID_q	$\leftrightarrow$	t_s
ID_q	$\leftrightarrow$	t_f
q_{code}	$\leftrightarrow$	k
q_{code}	$\leftrightarrow$	$\mathbf{X}$
q_{code}	$\leftrightarrow$	t_s
q_{code}	$\leftrightarrow$	t_f
k	$\leftrightarrow$	$\mathbf{X}$
k	$\leftrightarrow$	t_s
k	$\leftrightarrow$	t_f
$\mathbf{X}$	$\leftrightarrow$	t_s
$\mathbf{X}$	$\leftrightarrow$	t_f
t_s	$\leftrightarrow$	t_f

An input file corresponding to the connections in the graph is created. Tarjan's algorithm is applied to this input files.

Output:

The output shows that there are five sets of *SCC* as shown in Figure 7.24

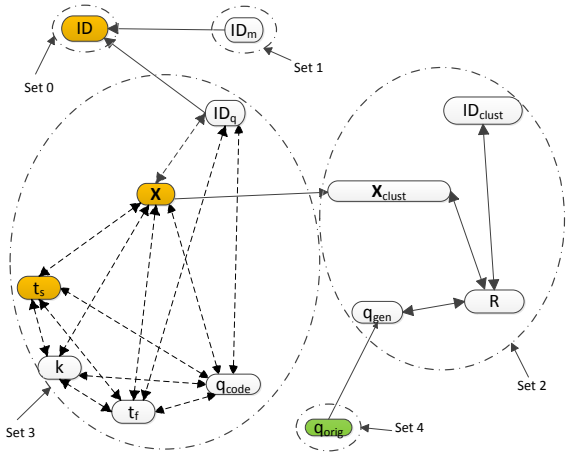


Figure 7.24: Output graph under "Attack on message string to CPS_L" condition

Set 0 consists of ID, Set 1 consists of ID_m and Set 2 consists q_{orig}. The other two sets: Set 2

contains parameters known to the LBSP and Set 3 contains ID_q, q_{code} , k, **X** , t_s and t_f . Since it is possible to decipher **X** and t_s, which is part of Set 3, from the knowledge of ID_q making it decipherable under the condition that the ID can be deciphered through the connection with ID_m or ID_q and hence marked with orange color. Since it is also possible to decipher ID through the weak connection with ID_q/ID_m, we indicate this parameter on the graph in orange color. Since q_{orig} is connected to Set 2, but the directionality is to Set 2, the q_{orig} is not decipherable from the knowledge of Set 2 and shown in green color. **As seen in the graph, it is not possible to establish connections against all four privacy parameter, it can be safely said that users privacy is preserved.**

Case 12: Message string to server CPS_L is attacked by an adversary (with ID_m and ID_q encrypted):

Input Graph:

Next case considered is when the attacker sniffs the message string to server CPS_L (q_{CPS_L}) with ID_m and ID_q encrypted. Removal of the connectivity associated with the unidirectional connection between ID_m/ID_q and ID from the input file of Case 11, an input file corresponding to the connections in the graph is created. Tarjan's algorithm is applied to this input files.

Output:

The output shows that there are three sets of *SCC* as shown in Figure 7.25

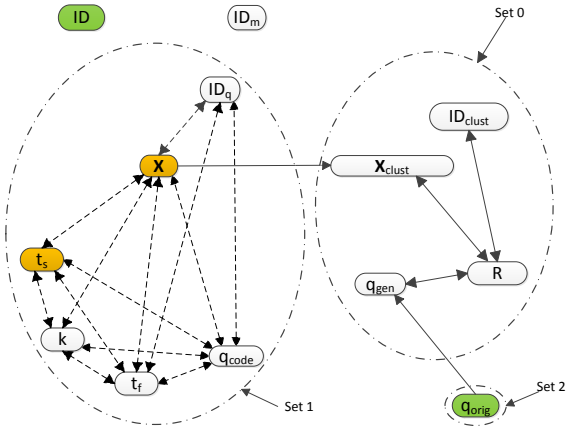


Figure 7.25: Output graph under "Attack on message string to CPS_L" condition with ID_m and ID_q encrypted

As can be seen from the graph, the ID (unconnected to any *SCC*) and q_{orig} (explanation same as in Case 11) are not decipherable from any *SCC* and hence depicted in green color. Since it

is possible to decipher $\mathbf{X}$ and t_s, which is part of Set 3, from the knowledge of ID_q making it decipherable under the condition that the ID can be deciphered through connection with ID_m or ID_q and hence marked with orange color.. **Since two out of four privacy parameters (**ID **and** q_{orig} **) are not decipherable, it can be said that privacy is protected if the message string to CPS_L server, with** ID_m **and** ID_q **encrypted, is sniffed by the adversary**.

Case 13: Message string from server CPS_L to server CPS_Q is attacked by an adversary (with ID_m **and** ID_q **unencrypted):**

Input Graph:

Next case considered is when the attacker sniffs the message string from server CPS_L to server CPS_Q. Apart from the basic edges of the graph from Case 1, bi-directional arrows are included to show the input connections between the parameters ID_q, q_{code}, $\mathbf{X}_{clust}$, t_s and t_f. which are parameters of the message string to the CPS_L server. Thus, the connections, in addition to those in Case 1 are as shown

Parameter	Connectivity	Parameter
ID_q	$\leftrightarrow$	q_{code}
ID_q	$\leftrightarrow$	$\mathbf{X}_{clust}$
ID_q	$\leftrightarrow$	t_s
ID_q	$\leftrightarrow$	t_f
q_{code}	$\leftrightarrow$	$\mathbf{X}_{clust}$
q_{code}	$\leftrightarrow$	t_s
q_{code}	$\leftrightarrow$	t_f
$\mathbf{X}_{clust}$	$\leftrightarrow$	t_s
$\mathbf{X}_{clust}$	$\leftrightarrow$	t_f
t_s	$\leftrightarrow$	t_f

An input file corresponding to the connections in the graph is created. Tarjan's algorithm is applied to this input files.

Output:

The output shows that there are five sets of *SCC* as shown in Figure 7.26

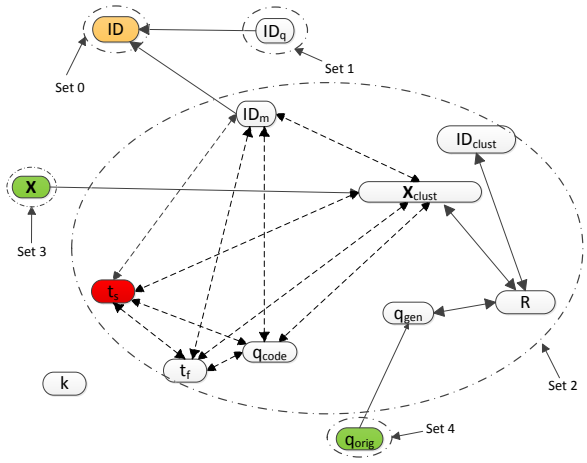


Figure 7.26: Output graph under "Attack on message string from CPS_L to CPS_Q" condition

Set 0 consists of ID, Set 1 consists of ID_m and Set 3 consists of $\mathbf{X}$ and Set 4 consists of q_{orig} The only other set:, Set 2, contains the privacy parameter t_s that can be deciphered since some of the other parameters that are accessible to the adversary ($\mathbf{X}_{clust}, ID_{clust}$ and q_{gen}) and so is depicted in red color. The ID can be deciphered through weak connection with ID_m or ID_q and hence marked with orange color. Since q_{orig} is connected to Set 2, but the directionality is to Set 2, the q_{orig} is not decipherable from the knowledge of Set 2 and shown in green color. Similar is the case with the location parameter $\mathbf{X}$. **As seen in the graph, it is possible to decipher the values of just two privacy parameters at most (ID and q_{orig}). Thus, it can be safely said that users privacy is preserved.**

Case 14: Message string between servers CPS_L and server CPS_Q is attacked by an adversary (with ID_m and ID_q encrypted):

Input Graph:

Next case considered is when the attacker sniffs the message string from server CPS_L to server CPS_Q with ID_m and ID_q encrypted . The only difference in the input file for this case from the input file of case 13 is the deletion of the two entries from the input file corresponding to the $ID_m \rightarrow ID$ and $ID_q \rightarrow ID$ connections.

An input file corresponding to the connections in the graph is created. Tarjan's algorithm is applied to this input files.

Output:

The output shows that there are three sets of *SCC* as shown in Figure 7.27

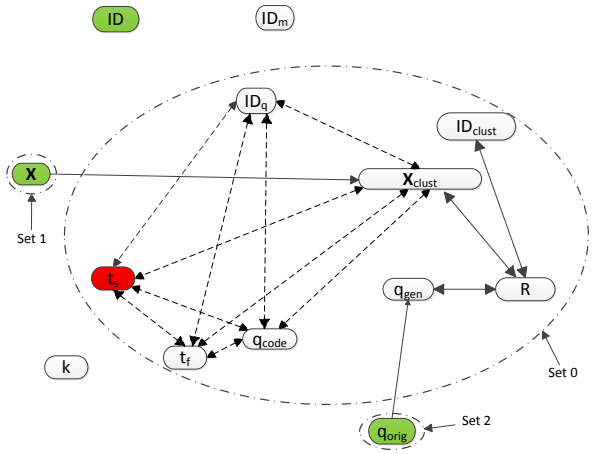


Figure 7.27: Output graph under "Attack on message string from CPS_L to CPS_Q" condition

Set 1 consists of **X** and Set 2 consists of q_{orig} The only other set:, Set 0, contains the privacy parameter t_s that can be deciphered since some of the other parameters that are accessible to the adversary ($\mathbf{X}_{clust}, ID_{clust}$ and q_{gen}) and so is depicted in red color. The privacy parameter ID is not decipherable since it is not part of any *SCC* and hence marked with green color. Since q_{orig} is connected to Set 0, but the directionality is to Set 0, the q_{orig} is not decipherable from the knowledge of Set 0 and shown in green color. Similar is the case with the location parameter **X**. **As seen in the graph, it is possible to decipher the values of just one privacy parameter (t_s) which does not cause a threat to the user. Thus, it can be safely said that users privacy is preserved.**

7.2.5 Set 5: Attack on the message strings with an attack on any one server

Case 15: Message string to server CPS_Q and the CPS_Q server is attacked simultaneously by an adversary (with ID_m and ID_q unencrypted):

Input graph:

This case essentially considers combination of Case 3 and Case 9. An input file combining the input files for these cases is created and the duplicate entries are removed. Tarjan's algorithm is applied to this input file.

Output:

The output shows that there are three sets of *SCC* as shown in Figure 7.28

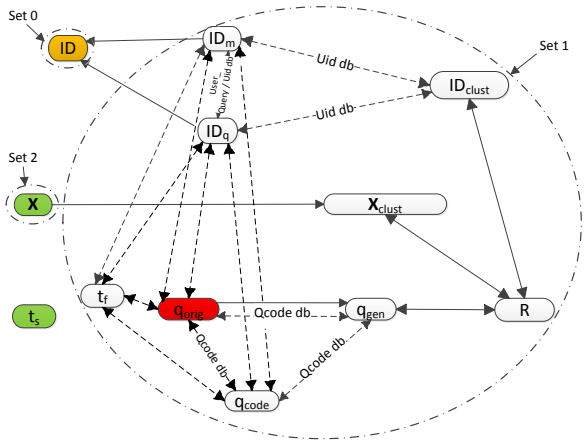


Figure 7.28: Output Graph under "Attack on message string to CPS_Q and the CPS_Q server"

Set 0 comprises of ID and Set 2 consists of **X**. The only other set:, Set 1, contains the privacy parameter t_s that can be deciphered since some of the other parameters that are accessible to the adversary ($\mathbf{X}_{clust}, ID_{clust}$ and q_{gen}) and so is depicted in red color. The privacy parameter ID is decipherable only through weak connection between ID_m or ID_q hence marked with orange color. Since location parameter **X** is connected to Set 1, but the directionality is to Set 1, the X parameter is not decipherable from the knowledge of Set 0 and shown in green color. Similar is the case with the . **As seen in the graph, it is not possible to decipher the values of just two privacy parameters (X and t_s) and hence does not cause a threat to the user. Thus, it can be safely said that users privacy is preserved.**

Case 16: Message string to server CPS_Q and the CPS_Q server is attacked simultaneously by an adversary (with ID_m and ID_q encrypted):

Input graph:

This case essentially considers combination of Case 4 and Case 10. An input file combining the input files for these cases is created and the duplicate entries are removed. Tarjan's algorithm is applied to this input file.

Output:

The output shows that there are just two sets of *SCC* as shown in Figure 7.29

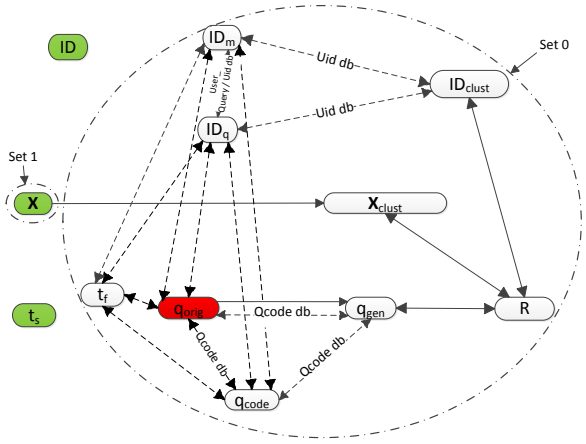


Figure 7.29: Output Graph under "Attack on message string to CPS_Q and the CPS_Q server" with ID_m and ID_q encrypted

As can be seen in the output graph, there are only two *SCC*s. Only the time privacy parameter $t_s$is revealed to the adversary and the parameters ID, **X** and t_s are not revealed and hence **it can be safely said that the privacy of the user is preserved under this condition.**

Case 17: Message string to server CPS_Q and the CPS_L server is attacked simultaneously by the adversary (with ID_m and ID_q unencrypted):

Input graph:

This case essentially considers combination of Case 5 and Case 9. An input file combining the input files for these cases is created and the duplicate entries are removed. Tarjan's algorithm is applied to this input file.

Output:

The output shows that there are just two sets of *SCC* as shown in Figure 7.30

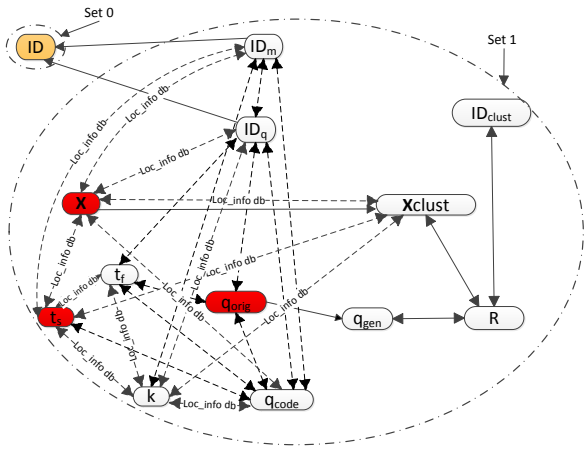


Figure 7.30: Output Graph under "Attack on message string to CPS_Q and the CPS_L server"

Note that this condition implies almost like the adversary being an universal observer and hence is most unsafe. As mentioned earlier, this framework does not guarantee protection from the universal observers. However, note that in this situation, the identity parameter ID (Set 0) is decipherable only through weak connection and not otherwise. This case is similar to the case of global adversary attack and so can be a threat to privacy of the user.

Case 18: Message string to server CPS_Q and the CPS_L server is attacked simultaneously by an adversary (with ID_m and ID_q encrypted):

Input graph:

This case essentially considers combination of Case 6 and Case 10. An input file combining the input files for these cases is created and the duplicate entries are removed. Tarjan's algorithm is applied to this input file.

Output:

The output shows that there is just one set of *SCC* as shown in Figure 7.31

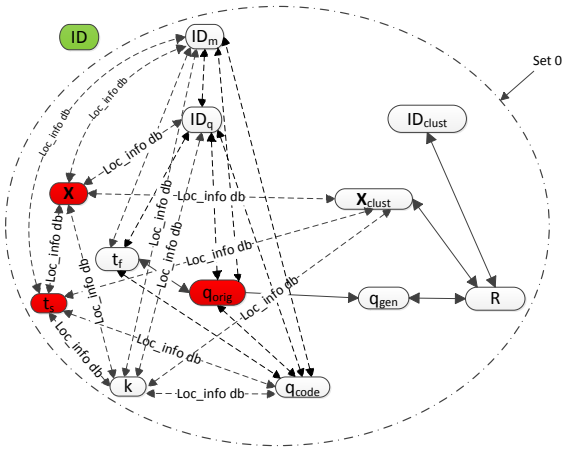


Figure 7.31: Output Graph under "Attack on message string to CPS_Q and the CPS_L server" with ID_m and ID_q encrypted

As specified in Case 17, this condition implies that the adversary is an universal observer and hence is most unsafe. However, note that in this situation when the unique request reference pair $< ID_m, ID_q >$is encrypted, the identity parameter ID (Set 0) is not decipherable. **In spite of an attack from a universal observer, the output graph indicates that the Identity parameter (ID) of the user is still protected.**

Case 19: Message string to server CPS_L and the CPS_Q server is attacked simultaneously by the adversary (with ID_m and ID_q unencrypted):

Input graph:

This case essentially considers a combination of Case 3 and Case 11. An input file combining the input files for these cases is created and the duplicate entries are removed. Tarjan's algorithm is applied to this input file.

Output:

The output shows that there are two sets of *SCC* as shown in Figure 7.32

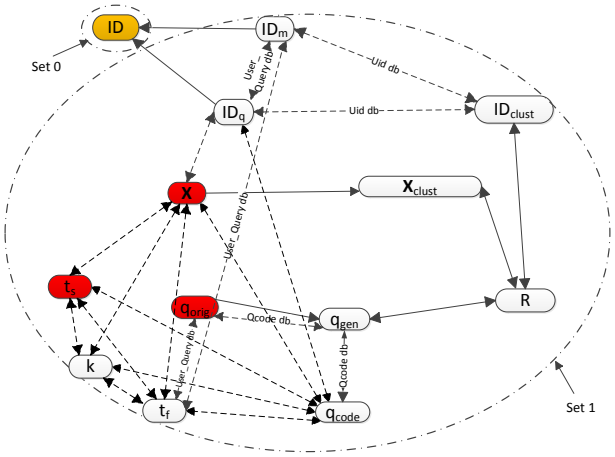


Figure 7.32: Output Graph under "Attack on message string to CPS_L and the CPS_Q server"

This case is also similar to Case 17 and as can be seen in the output graph, the Identity parameter can only be protected weakly (orange color) since there exists a weak connection between unique service request reference pair $< ID_m, ID_q >$.

Case 20: Message string to server CPS_L and the CPS_Q server is attacked simultaneously by an adversary (with ID_m and ID_q encrypted):

Input graph:

This case essentially considers a combination of Case 4 and Case 12. An input file combining the input files for these cases is created and the duplicate entries are removed. Tarjan's algorithm is applied to this input file.

Output:

The output shows that there is just one set of *SCC* as shown in Figure 7.33

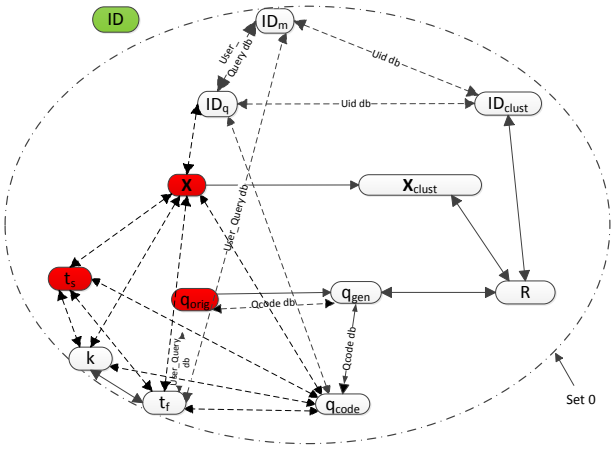


Figure 7.33: Output Graph under "Attack on message string to CPS_L and the CPS_L server" with ID_m and ID_q encrypted

As specified in Case 19, this condition implies that the adversary is an universal observer and hence is most unsafe. However, note that in this situation when the unique request reference pair $< ID_m, ID_q >$is encrypted, the identity parameter ID (Set 0) is not decipherable. **In spite of an attack from a universal observer, the output graph indicates that the Identity parameter (ID) of the user is still protected.**

Case 21: Message string to server CPS_L and the CPS_L server is attacked simultaneously by the adversary (with ID_m and ID_q unencrypted):

Input graph:

This case essentially considers a combination of Case 5 and Case 11. An input file combining the input files for these cases is created and the duplicate entries are removed. Tarjan's algorithm is applied to this input file.

Output:

The output shows that there are four sets of *SCC* as shown in Figure 7.34

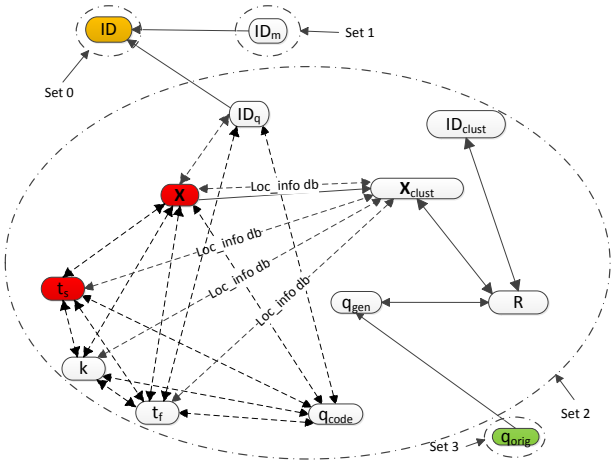


Figure 7.34: Output Graph under "Attack on message string to CPS_L and the CPS_L server"

Set 0 comprises of ID, Set 1 consists of ID_m and Set 3 comprises of the original query parameter q_{orig}. Set 2, contains the privacy parameters $\mathbf{X}$ and t_s that can be deciphered since some of the other parameters that are accessible to the adversary ($\mathbf{X}_{clust}, ID_{clust}$ and q_{gen}) and so is depicted in red color. The privacy parameter ID is decipherable only through weak connection between ID_m or ID_q hence marked with orange color. Since query parameter q_{orig} is connected to Set 2, but the directionality is to Set 2, the q_{orig} parameter is not decipherable from the knowledge of Set 2 and shown in green color. **As seen in the graph, it is possible to decipher the values of just two privacy parameters ($\mathbf{X}$ and t_s). The ID parameter is weakly identified. Thus, it can be said that user's privacy is protected.**

Case 22: Message string to server CPS_L and the CPS_L server is attacked simultaneously by the adversary (with ID_m and ID_q encrypted):

Input graph:

This case essentially considers a combination of Case 6 and Case 12. An input file combining the input files for these cases is created and the duplicate entries are removed. Tarjan's algorithm is applied to this input file.

Output:

The output shows that there is just one set of *SCC* as shown in Figure 7.35

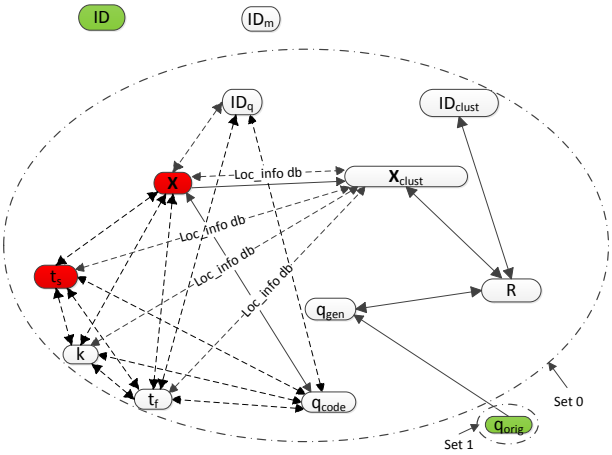


Figure 7.35: Output Graph under "Attack on message string to CPS_L and the CPS_L server" with ID_m and ID_q encrypted

The graph shows that two privacy parameters: ID and q_{orig} are not decipherable and hence shown in green color. The reasons are similar to the explanations given in similar situations discussed earlier. The parameters Xand t_s are exposed to the adversary, but as explained earlier, just knowing the location and the time of query (X andt_s) does not associate it to a particular user and **hence it can be safely said that privacy is protected.**

Case 23: Message string between servers CPS_L and CPS_Q and the CPS_Q server is simultaneously attacked by the adversary (with ID_m and ID_q unencrypted):

Input graph:

This case essentially considers a combination of Case 3 and Case 13. An input file combining the input files for these cases is created and the duplicate entries are removed. Tarjan's algorithm is applied to this input file.

Output:

The output shows that there are three sets of *SCC* as shown in Figure 7.36

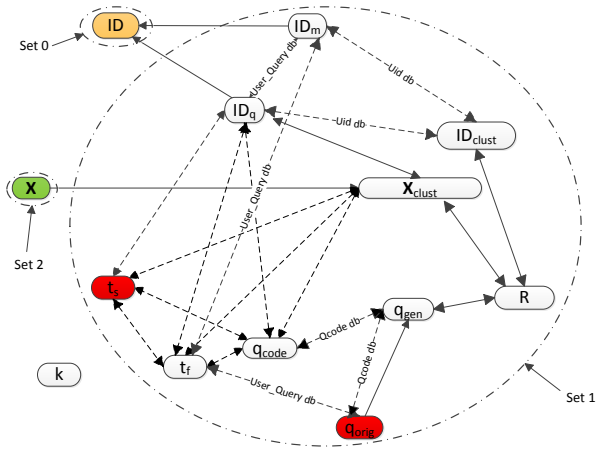


Figure 7.36: Output Graph under "Attack on message string between CPS_L and CPS_Q and the CPS_Q server"

As can be seen in the output graph, the Identity parameter can only be protected weakly (orange color) since there exists a weak connection between unique service request reference pair $< ID_m, ID_q >$. The location parameter **X** has a weak connection with *SCC* Set 1, but the directionality favors the location parameter and hence is marked with green color. The other two privacy parameters t_s and q_{orig} are however decipherable since they form a part of *SCC* that includes parameters that are available to the adversary. **Since at least one parameter (X) is not decipherable, it can be said that the user's privacy is reasonably protected**.

Case 24: Message string between servers CPS_L and CPS_Q and the CPS_Q server is simultaneously attacked by the adversary (with ID_m and ID_q encrypted):

Input graph:

This case essentially considers a combination of Case 4 and Case 14. An input file combining the input files for these cases is created and the duplicate entries are removed. Tarjan's algorithm is applied to this input file.

Output:

The output shows that there are just two sets of *SCC* as shown in Figure 7.37

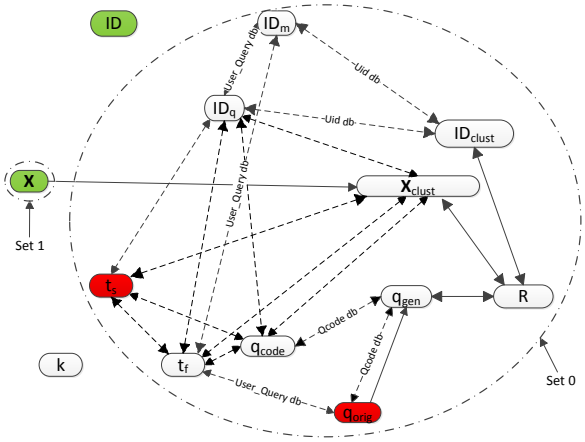


Figure 7.37: Output Graph under "Attack on message string between CPS_L and CPS_Q and the CPS_Q server" with ID_m and ID_q encrypted

The output graph indicates that the Identity parameter and the location parameter are not decipherable and hence marked in green color. The other two privacy parameters t_s and q_{orig} are however decipherable since they form a part of *SCC* that include the parameters that are available to the adversary. **Since two parameters (ID and X) are not decipherable, it can be said that the user's privacy is reasonably protected**.

Case 25: Message string between servers CPS_L and CPS_Q and the CPS_L server is attacked simultaneously by an adversary (with ID_m and ID_q unencrypted):

Input graph:

This case essentially considers a combination of Case 5 and Case 13. An input file combining the input files for these cases is created and the duplicate entries are removed. Tarjan's algorithm is applied to this input file.

Output:

The output shows that there are four sets of *SCC* as shown in Figure 7.38

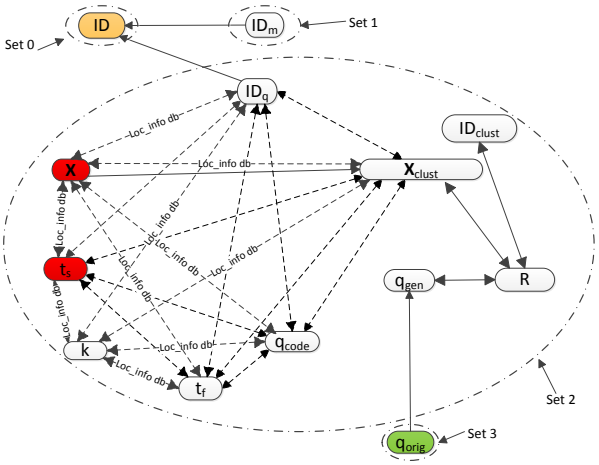


Figure 7.38: Output Graph under "Attack on message string between CPS_L and CPS_Q and the CPS_L server"

Set 0 comprises of ID, Set 1 consists of ID_m and Set 3 comprises of the original query parameter q_{orig}. The privacy parameter ID is decipherable only through weak connection between ID_m or ID_q hence marked with orange color. Set 2, contains the privacy parameters **X** and t_s that can be deciphered since some of the other parameters that are accessible to the adversary ($\mathbf{X}_{clust}, ID_{clust}$ and q_{gen}) and so is depicted in red color. Since query parameter q_{orig} is connected to Set 2, but the directionality is to Set 2, the q_{orig} parameter is not decipherable from the knowledge of Set 2 and shown in green color. **As seen in the graph, it is possible to decipher the values of just two privacy parameters. The ID parameter is weakly identified. Thus, it can be said that user's privacy is protected.**

Case 26: Message string between servers CPS_L and CPS_Q and the CPS_L server is attacked by adversary (with ID_m and ID_q encrypted):

Input graph:

This case essentially considers a combination of Case 6 and Case 14. An input file combining the input files for these cases is created and the duplicate entries are removed. Tarjan's algorithm is applied to this input file.

Output:

The output shows that there are two sets of *SCC* as shown in Figure 7.39

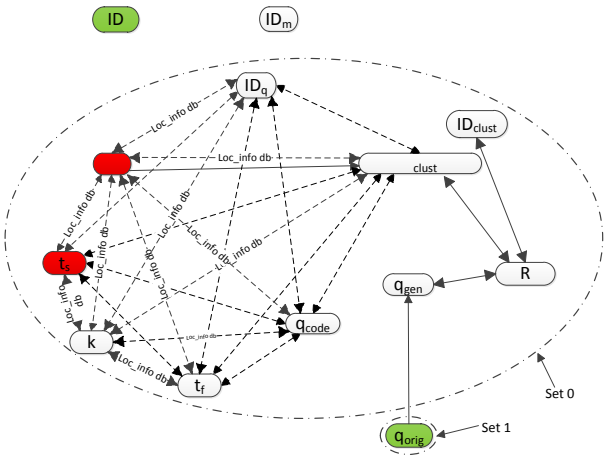


Figure 7.39: Output Graph under "Attack on message string between CPS_L and CPS_Q and the CPS_L server" with ID_m and ID_q encrypted

The graph shows that two privacy parameters: ID and q_{orig} are not decipherable and hence shown in green color. The reasons are similar to the explanations given in similar situations discussed earlier. The parameters **X**and $t_s$are exposed to the adversary, but as explained earlier, just knowing the location and the time of query does not associate it to a particular user (due to k anonymity) and **hence it can be safely said that privacy is protected.**

Table 7.3 gives a comprehensive overview of the various cases discussed.

Table 7.3: Comprehensive overview of various cases

Level of protection	Cases	Privacy Parameters			Type of Attack
		Protected	Conditionally Protected	Exposed	
High	1	**X**, t_s, q_{orig}	ID		No Attack
	2	ID, **X**, t_s, q_{orig}			No Attack
	4	ID, **X**, t_s		q_{orig}	DB Attack
	10	ID, **X**, t_s	q_{orig}		Comm. Attack
	14	ID, **X**, q_{orig}		t_s	Comm. Attack
	16	ID, **X**, t_s		q_{orig}	Comm. Attack + DB Attack
Medium	3	**X**, t_s	ID	q_{orig}	DB Attack
	6	ID, q_{orig}		**X**, t_s	DB Attack
	9	**X**, t_s	ID, q_{orig}		Comm. Attack
	11	q_{orig}	ID, **X**, t_s		Comm. Attack
	12	ID, q_{orig}	**X**, t_s		Comm. Attack
	13	**X**, q_{orig}	ID	t_s	Comm. Attack
	15	**X**, t_s	ID	q_{orig}	Comm. Attack + DB Attack
	22	ID , q_{orig}		**X**, t_s	Comm. Attack + DB Attack
	24	ID, **X**		t_s, q_{orig}	Comm. Attack + DB Attack
	26	ID, q_{orig}		**X**, t_s	Comm. Attack + DB Attack
Low	5	q_{orig}	ID	**X**, t_s	DB Attack
	7		ID	**X**, t_s,q_{orig}	DB Attack
	8	ID		**X**, t_s,q_{orig}	DB Attack
	17		ID	**X**, t_s,q_{orig}	Comm. Attack + DB Attack
	18	ID		**X**, t_s,q_{orig}	Comm. Attack+DB Attack
	19		ID	**X**, t_s,q_{orig}	Comm. Attack+DB Attack
	20	ID		**X**, t_s,q_{orig}	Comm. Attack + DB Attack
	21	q_{orig}	ID	**X**, t_s	Comm. Attack+DB Attack
	23	**X**	ID	t_s,q_{orig}	Comm. Attack + DB Attack
	25	q_{orig}	ID	**X**, t_s	Comm. Attack + DB Attack

CHAPTER 8

GENERIC PRIVACY PROTECTION COMPARISON

In this section, we prove that the privacy protection offered by the CLOPRO framework is better than the privacy offered by the existing solutions. To prove that the privacy protection is offered by the various existing techniques under an attack from the adversary, we need to evaluate the probability of rediscovery of the parameters based on the apriori knowledge of the parameters to the adversary. We then show that probability of rediscovery in the case of CLOPRO framework is the least.

8.1 Privacy Protection Comparison

The highest form of protection to privacy is provided when all of the parameters revealing the user are decipherable by the adversary. Any service request comprises of the parameters: the identity of the user, the location of the user at the time of the service request, the time of request and the actual query. As mentioned earlier, we combine the two parameters location and time of query as the context of the user. So when we evaluate the level of privacy, hence forth, we will just consider them as three parameters. The existing solutions offer privacy to either one [7, 11-13, 15-32, 34-42] or two [15, 34, 43] of the three privacy protecting parameters. The CLOPRO framework provides privacy protection to all the three components. We now demonstrate with a proposition that the CLOPRO framework provides highest degree of privacy protection.

We assume that an adversary may have some knowledge, especially since we consider that the LBSP can also be an adversary. To find out the level of privacy offered by the solution, we have to find the probability of an adversary rediscovering the privacy parameters of interest based on the knowledge of certain other parameters. We calculate the probability of rediscovery of status of any one variable, given the knowledge of the status of other parameters. We can thus apply the Bayesian Chain rule in this situation.

Proposition 2

The degree of privacy protection in CLOPRO framework is better than that provided by the existing approaches since the probability of rediscovery of all three components is less than the probability of rediscovery of any one or two privacy protection parameters in the service request process.

Proof of Proposition 2

Let us first define three bistate variables (x, y, and z) which would represent the state of data privacy as:

1. x := status of identity. $x = T$ or F such that $x = T$ means individual identity is protected.

2. y := status of query. $y = T$ or F such that $y = T$ means individual's query is anonymized and hence protected.

3. z := status of location. $z = T$ or F such that $z = T$ means individual's location is anonymized and hence protected.

Let us now define $p(x_i)$ as the probability that x is at state i ($i \in \{T, F\}$) . As x, y, and z are independent of each other, we can compute the conditional probability

$$p(x_i) = \sum_k \sum_j p(x_i|_{y_j,z_k}) \times p(y_j|_{z_k}) \times p(z_k) \tag{8.1}$$

where $p(x_i|_{y_j,z_k})$ represents the probability the x is at state i given that y is at state j and z is at state k. Similarly, $p(y_j|_{z_k})$ represents the probability that y is at state j given z is at state k. To compute the probability that identity is compromised,

$$\begin{aligned} p(x = F) = (&p(x = F|_{y=T,z=T}) \times p(y = T|_{z=T}) \times p(z = T) \\ &+ p(x = F|_{y=T,z=F}) \times p(y = T|_{z=F}) \times p(z = F) \\ &+ p(x = F|_{y=F,z=T}) \times p(y = F|_{z=T}) \times p(z = T) \\ &+ p(x = F|_{y=F,z=F}) \times p(y = F|_{z=F}) \times p(z = F) \end{aligned} \tag{8.2}$$

The complete privacy breach will be when all the three components are compromised. The probability for such a worst case will be given by$p_{worstcase} = P_{rediscovery}(x = F, y = F, z = F)$ as

$$p_{worstcase} = p(x = F|_{y=F,z=F}) \times p(y = F|_{z=F}) \times p(z = F) \tag{8.3}$$

Based on the above equations, it can be easily deduced that

$$p_{worstcase} \leq p(x = F) \tag{8.4}$$

which means the probability that only the identity is compromised are higher than the complete loss of private information. Extending the same logic, one can deduce that

$$P_{rediscovery}(x = F, y = F, z = F) < P_{rediscovery}(x = F, y = F) < P_{rediscovery}(x = F) \tag{8.5}$$

Note that the two components may be $(x = F, z = F)$ or $(y = F, z = F)$ instead of $(x = F, y = F)$ and the single component could be $(y = F)$or $(z = F)$ instead of $(x = F)$.
Thus, we prove that the CLOPRO framework that comprehensively protects the privacy of the user while using location based services provides the best protection.

CHAPTER 9

EXPERIMENTAL SETUP AND PRELIMINARY RESULTS

9.1 Experimental Setup

The experiments are run on a standard desktop machine with Intel core i-3 processor, 4 GB RAM and a 32 bit Windows -7 Operating System. The environment used is Visual Studio. Three servers (CPS_L, CPS_Q, LBSP) are setup using MSDN's Web Communication Foundation (WCF), a framework for building service-oriented applications since it is necessary to send the service request as asynchronous messages from one service endpoint, hosted on Internet Information Services (IIS) , to another. The endpoint was the client or the mobile device that requests data from a service endpoint. The service requests were sent as XML.The framework used for the development was ASP.NET and coding was done in C#.

The main modules to simulate the operations of the CLOPRO framework (Figure 9.1) are:

- ♢ **Qcode Generator** - Generation of q_{code} for various LBS applications. This operation is performed offline whenever a new application is desired. A subset of this database is sent to the user's mobile device based on the applications the user wishes to use.
- ♢ **Context Cloaker** - This is a module on mobile device of the users for registration, creation and segregation of the service request. It also triggers the process of anonymization and display of the results to the user.
- ♢ **CPS_L Service** - This module is responsible for transforming the location of the users, disbursing the results to the user. These module works in the background.
- ♢ **CPS_Q Service** - This module performs multiple tasks. The first task is the registration of the user and the LBSP. The module rebuilds the query and obtains the results from the LBSP. These results are processed and the reconstructed results are sent to the user through the CPS_ service. This module also works in the background.

♢ **Map** - This module executes the process of anonymization , reporting of timeout requests, and generation of the cluster map. The module also shows the users whose desired response time is less than the processing time required due to high traffic at the time of service request.

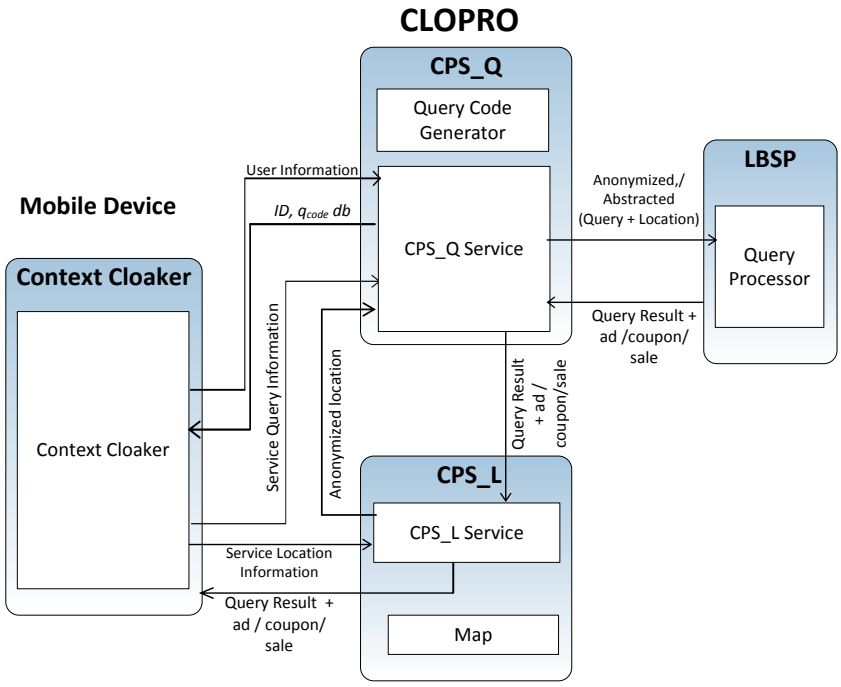


Figure 9.1: Diagram of implementation of the CLOPRO framework

9.1.1 Qcode Generator

Operation of the Qcode generation is performed offline by the Query Anonymizing server CPS_Q. The design of the generator is such that the q_{code} can be generated for any type of application. The module designed for this operation is generic. The operation consists of three steps:

♢ **Enter the Application Name**: To generate q_{code}, the name of the application has to be provided. A new unique code based on the first alphabet of the application name concatenated with a two digit number (based on the last two digit code that has been used in the past). For example, if the application name is Event Finder, and this is the first application with the name beginning with 'E" then the code generated is 'E01". This code is used for further processing and creating various q_{code}s for that application.

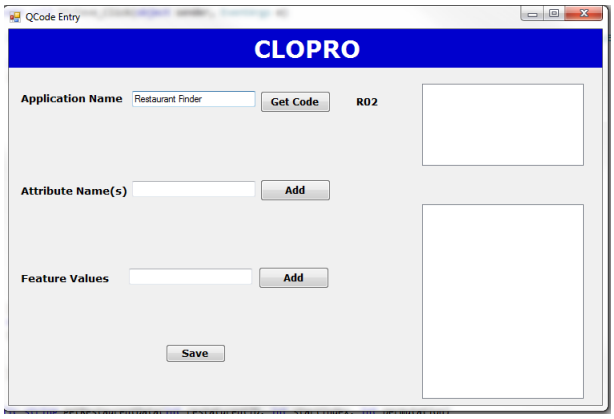


Figure 9.2: Qcode generator, application code generation

♢ **Enter the names of the Features of the application:** After the code for the application is obtained, next step is to specify various features of application. These are the features that would be used for creating the generic queries for the applications. Each of the entered features are shown in the box in top right corner. In this model, we have used creation of generic queries by combination of up to three of these features.

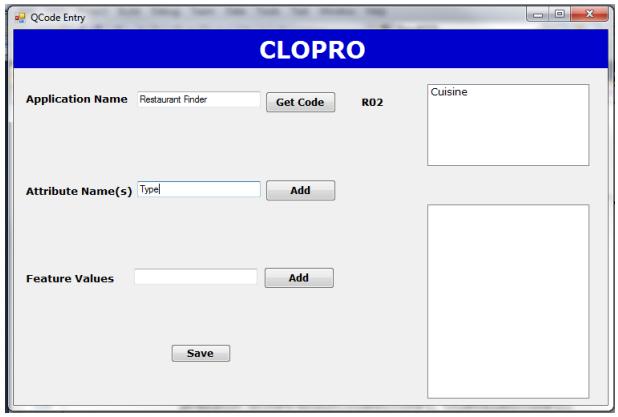


Figure 9.3: Qcode generation, feature entry

♢ **Enter the names of the values of the Leaf Features for each Feature:** After all the feature names are entered, the leaf feature values for each of the features is entered. As these leaf feature values are entered,they are displayed in the box at the bottom. Once all the leaf features for each of the features is entered, this can be saved to the database. Figure 9.2, Figure 9.3, Figure 9.4 shows the snap shots of the input screens for Qcode generation. So for example, in the Restaurant Finder, the names of the features specified

are 'Cuisine' and 'Type', and we have specified 'Indian', Italian', 'Mexican' and 'Chinese' as feature values for 'Cuisine' and 'Fastfood' and 'Sit_in' as feature values for 'Type' then a total 81 codes will be generated for various combinations. The initial six codes will be for each of feature value and the remaining will be disjunctive and conjunctive combinations of these features.

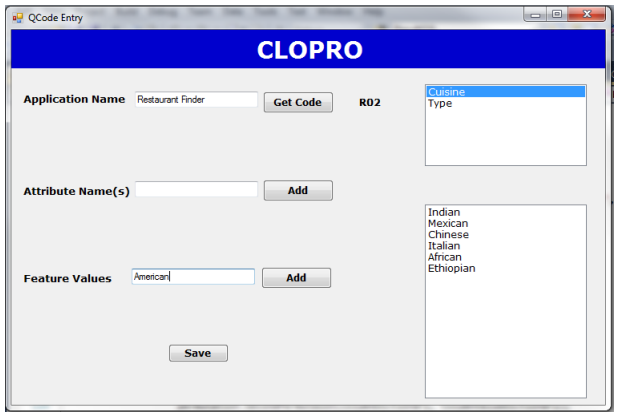


Figure 9.4: Qcode generation, leaf feature value entry

9.1.2 Context Cloaker

This module is resident on the mobile device of the user. This module, resident on the mobile device, is used to communicate with the CLOPRO framework, once for the registration of the user and the LBSP and repeatedly for making the service requests. The main input screen allows the user to specify if the user is a new user or the user is a registered user as shown in Figure 9.5.

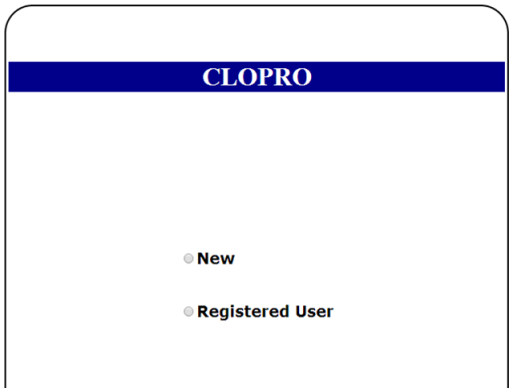
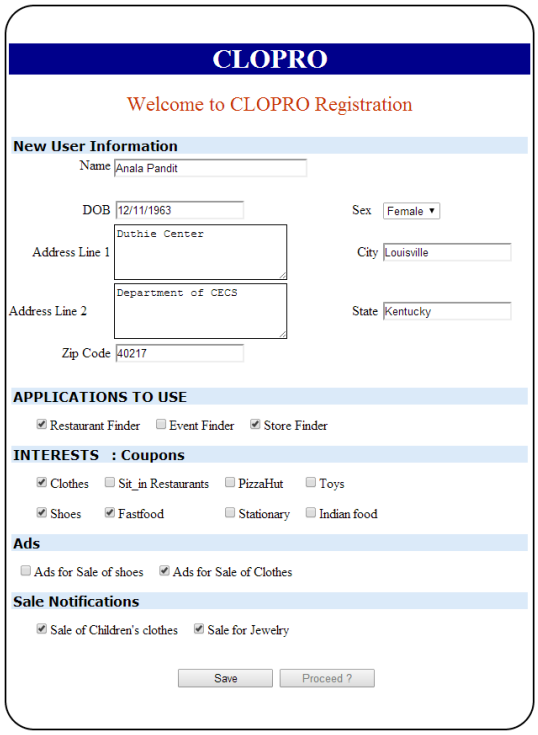
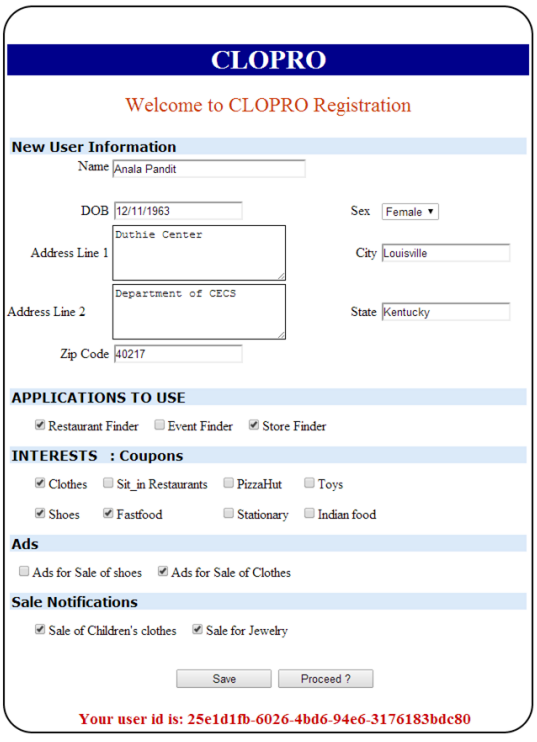


Figure 9.5: CLOPRO Main Page

During the process of registration in our sample application, Restaurant Finder, the user provides demographic information, the applications the user is desirous of using and the ads / coupons / sales the user is interested in receiving from the service provider. Figure 9.6 (a) shows the details of the information and Figure 9.6 (b) shows the user id generated after the process of registration. Once the user clicks on 'Save', a new unique id (ID) is generated for the user and displayed at the bottom of the page. This 'id' is appended with random numbers generated to create the unique request reference pair. If the user clicks on 'Proceed?' button, the main page is displayed again.



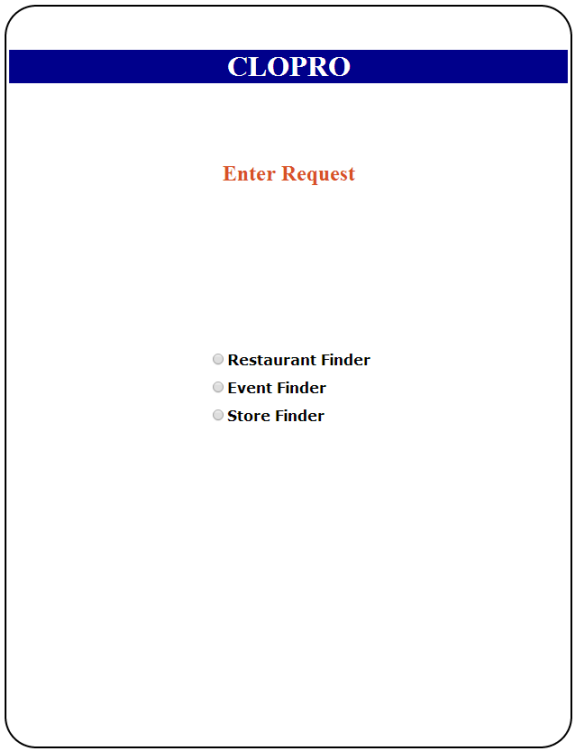
(a) User Registration



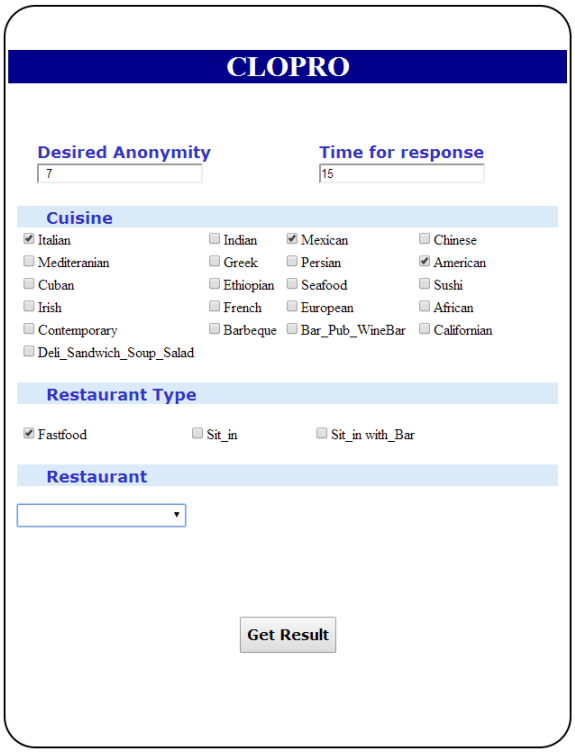
(b) Unique User_id

Figure 9.6: Context Cloaker Registration

On the main page, when the user selects the registered user, the user needs to select the application as shown in Figure 9.7a and enter the details regarding the desired anonymity level, the desired response time and the actual query as shown Figure 9.7b.



(a) Enter Application Name



(b) Enter Service Request

Figure 9.7: Service Request Entry

When the user clicks on the 'Show Message' button, an XML message based on the service request is generated and is appended to the queue of service requests as shown in Figure.

```
<Message>
 <userip>162.100.16.17</userip>
 <uid>6309bb1b-1e17-4bab-a3d5-1ac792e8a5fb_0_48938</uid>
 <qid>6309bb1b-1e17-4bab-a3d5-1ac792e8a5fb_1_48938</qid>
 <qcode>R41</qcode>
 <qorig>Cuisine ='Persian' or Cuisine ='Cuban' and Type ='Fastfood' or Type ='Sit_in'</qorig>
 <startTime>10-Feb-14 12:37:26 AM</startTime>
 <deltaT>15</deltaT>
 <K>5</K>
 <lat>38.213083</lat>
 <lng>-85.760835</lng>
</Message>
```

Figure 9.8: Snapshot of XML message generated

Also, the process of the *Location Anonymization* is invoked. After completion of the process of anonymizing, generic query generation and reconstruction of the result set, the CLOPRO framework responds back with the results of the service request as shown in Figure 9.9.

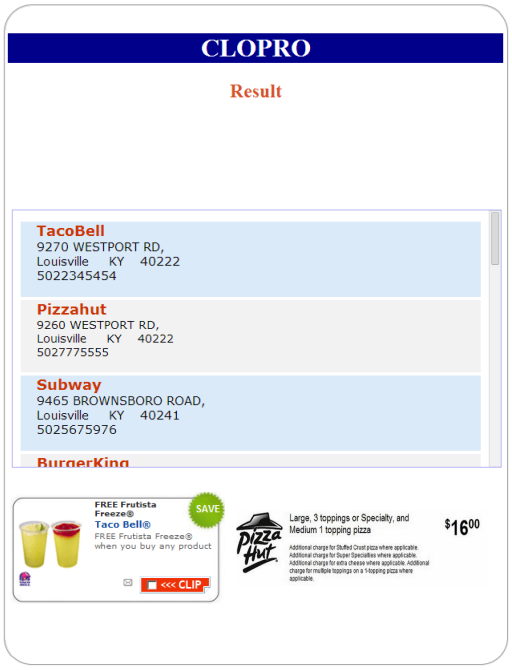


Figure 9.9: Service Request Result

In case the processing time exceeds the desired response time, a message as shown in Figure 9.10 appears on the device, providing the (k, T) anonymity property with a request to change.

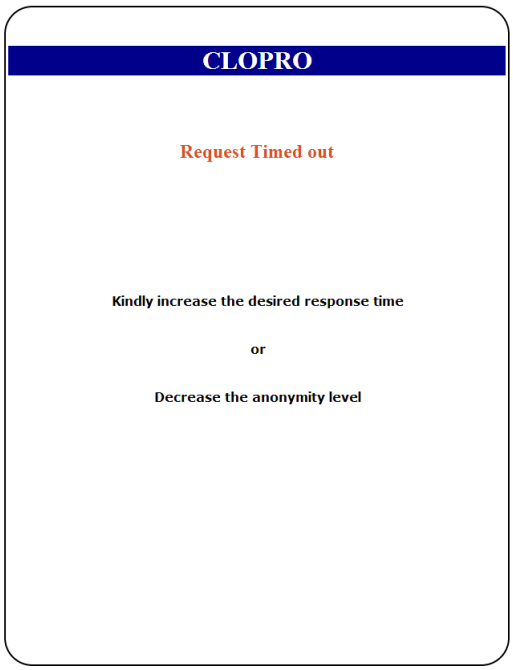


Figure 9.10: Request timed out screen

9.1.3 Map

For the purpose of evaluating the clusters, we have implemented a module MAP which allows one to view the clusters created in the process of *Location Anonymization* and to view the requests that have timed out. Currently, the maximum radius of the cluster is defined to be 5 km radius. By defining this radius, we are ensuring $< k, s >$ privacy requirement. The visual maps can be used to determine a threshold radius. The radius can be increased if it is found that there are too few elements in the cluster. If there are too many users in one cluster, over a period of time, this radius can be reduced and separate clusters can be created. The input screen is as shown in Figure 9.11. Once the anonymization process is completed, there are two types of operations possible.

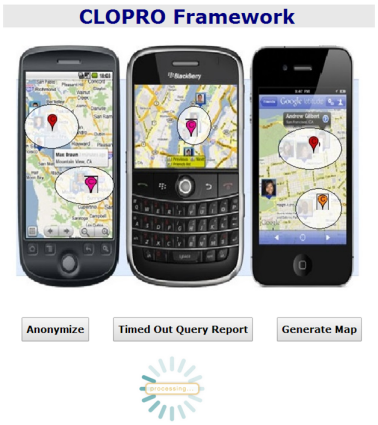


Figure 9.11: Input for Processing maps

The administrator can view the map indicating the clusters and the cluster centroids as shown in the Figure 9.12. Each of the points in the cluster is indicated by the pins of the same color. The cluster centroids are indicated using the pin with a 'C' in the center. As can be seen in the figure of the map 9.12, the cluster centroid is always different from the original points.

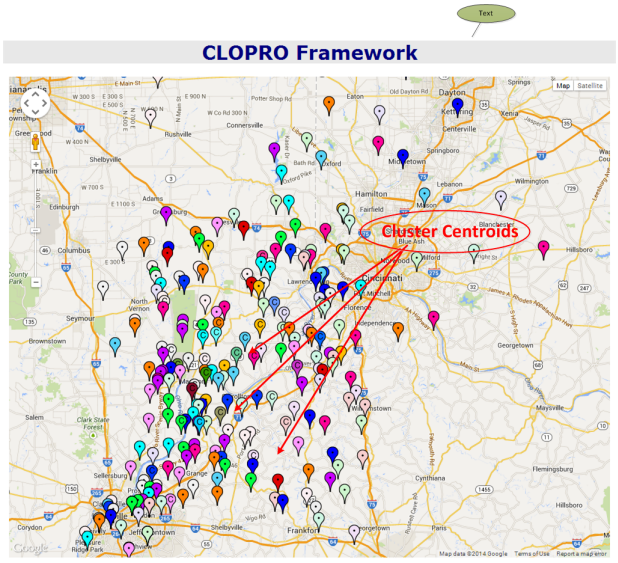


Figure 9.12: Map showing clusters

Although the user gets a notification of the request timed out, the administrator can also view the table that shows the service requests that have timed out along with the times taken for the process of anonymization as is seen in Figure . This report can be used for identifying the number of service requests that have not been received the result due to insufficient time specified by the user.

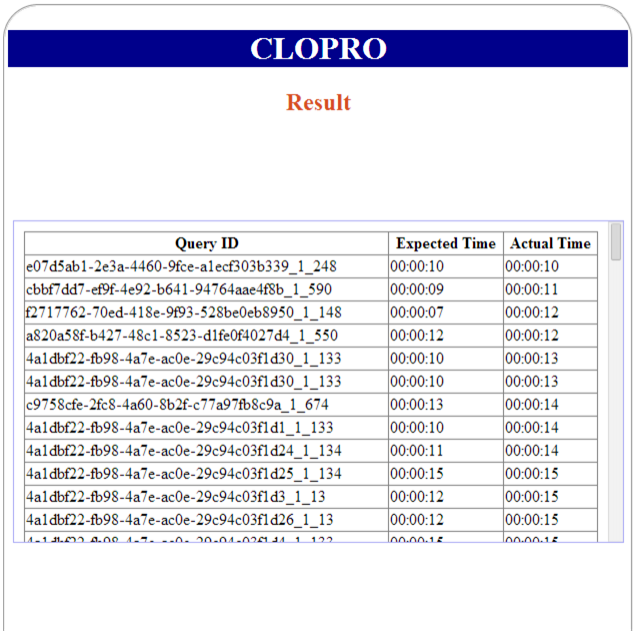


Figure 9.13: Timed Out Request Report

There were two types of experiments conducted to measure the performance of the system:

♢ **Varying number of service requests initiated in a time frame:** In this setup, we used synthetically generated requests within a certain time frame with varying number of users. We used number of users as: 10, 25, 50, 75, 100, 120. The response time is measured as the time from when the XML file, which holds the messages, is read to the time the result is generated for each service request. The limitation on the number of users is due to the constraints of the server's processing power. In this setup, each request had a different '*k*'. i.e. "***personalized anonymization***" for each request. The experiment was run using 10 fold cross validation. The average response time and the standard deviation is calculated. The results are presented below in Table 9.1

Table 9.1: Number of users v/s Response Time (secs)

Sr. No.	Number of Users	Av. Response Time(sec)	Std. Deviation
1	10	3.0943	0.3682
2	25	4.9718	0.7775
3	50	5.1103	0.9722
4	75	6.1773	1.0112
5	100	8.9456	0.6592
6	120	10.4479	0.7478

The graph of 'Number of users v/s the Average Response time' for receiving the response to the

service request is shown in Figure 9.14. As expected, the response time increases linearly with the number of users. With a server of higher processing power, this time can be minimized.

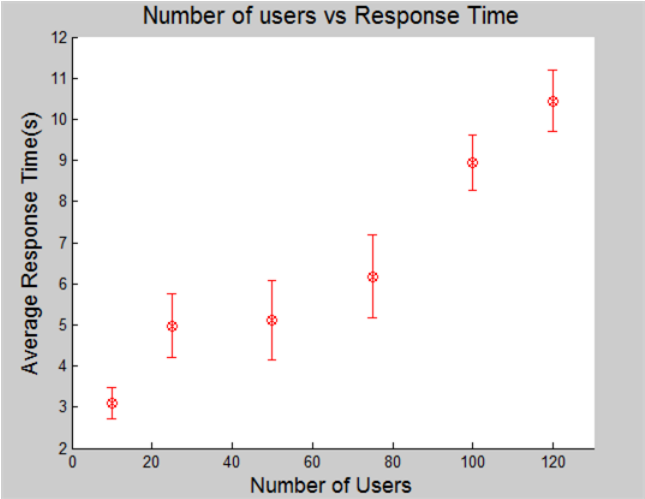


Figure 9.14: Variation in Response Time with number of users

♢ **Fixed number of service requests but varying *'k'*:** During this setup, the number of service requests was kept constant at 50. The value of *'k'* was varied. The values chosen were: 3, 5, 8, 10. Again the limitation is due to the processing power and the RAM of the server. Since the algorithm generates dummy requests to satisfy the *'k'* for every user, even though we have specified number of service requests as 50, there can be as many as 450 requests that the server processes, if there are very few requests in a single cluster. The results are shown in Table 9.2 and the variation is shown in Figure 9.15

Table 9.2: Variation in Response time with *'k'*

Sr. No.	k	Av. Response Time (secs)	Std. Deviation
1	3	4.6905	0.2815
2	5	6.3582	0.3369
3	8	9.8788	0.4297
4	10	10.2275	0.2678

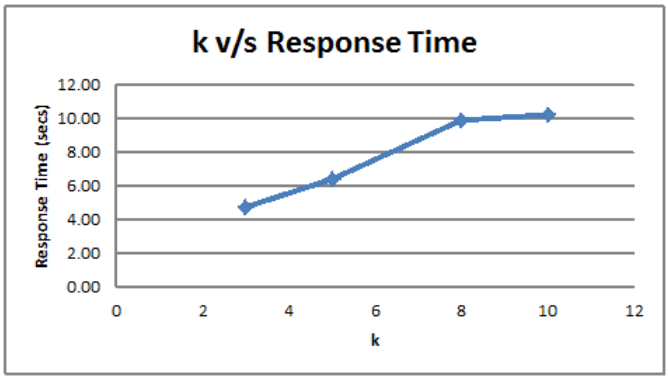


Figure 9.15: Variation in Response Time with variation in '*k*'

As seen in Figure 9.15 the response time increases as the server requires to process more number of requests to satisfy a larger '*k*' as expected. The standard deviation is low, but the time required is more than expected. However, the increase in the time is due to large number of dummy requests that were generated due to the quality of synthetic data. We believe that the results can be improved significantly with better dataset in both the cases.

CHAPTER 10

CONCLUSION AND FUTURE WORK

10.1 Conclusion

In this dissertation, we considered the complex task of designing a comprehensive framework to protect the privacy of the user while protecting the revenue model of the LBSP. The framework facilitates the user to use the LBSs and receive the additional services like obtaining the advertisements / coupons / sale notifications of desired services while protecting their privacy. The major issue with the solutions that protect the privacy of the users is that the LBSP's are starved of the revenue as the private information that is sent to the LBSP is falsified / altered using various techniques to protect the privacy.

The CLOPRO framework makes following novel and unique contributions:

• The development of a distributed, INTEGRATED framework preserves the *Identity* privacy, the *Context* privacy and the *Query* privacy while preserving the revenue model of the LBSP.

• The framework provides $< k, s >$ and $< k, T >$ anonymity but employing a simple clustering technique to cluster similar queries from users in a particular location area at specific time instance window. Since multiple queries are presented as a single query to the service provider reducing the load of the service provider. This improves the Quality of Service (QoS) of the service provider as large number of users are utilizing LBSs.

• The CLOPRO framework achieves query generalization process **offline**, improving the query response time. By reducing the granularity of the query, *Query* privacy protection is achieved.

• In the proposed framework CLOPRO, the technique of **caching** results improves the service request response time in repeat queries .

We designed and implemented a novel CLOPRO framework for Privacy Protection in Context Aware Systems using two non-colluding servers. We used a distributed architecture to ensure that a single point of attack that can breach the privacy is avoided. The system is simulated

on the ASP.NET 4.0 platform. The experimental setup consists of three web services CPS_L, CPS_Q, and LBSP. The Context Cloaker module of the CLOPRO framework, that resides on the mobile device is simulated as a separate web application. The web servers are simulated in the Microsoft's WCF framework. The process of Qcode generation (query anonymization module) is performed off line and a separate Window's application is developed for the Qcode generator.

We use the concept of anonymity in a crowd to provide protection. The service requests of various users are clustered and presented to the LBSP as a single request. We cluster the requests with similar requests initiated within a specific distance and a certain time window. Thus, we can say that our framework also satisfies $< k, s >$ and $< k, T >$ anonymity. Since the process of clustering of the user service requests This objective was achieved by designing a *Split Clustering Anonymization Algorithm* for the location and the *Query* privacy where we used distance based clustering techniques to provide the privacy to the user of Context Aware System. The implementation has been done as a proof of concept. In this implementation, the Identity Anonymity has been achieved through use of Unique Request Reference pair that is encrypted to ensure security. We use a simple distance based clustering mechanism over a specific distance for providing *Location Anonymity* and *Query Abstraction* technique for providing Query Anonymity.

We have proposed a formal definition of privacy and the attack. We further identified and formalized that the privacy is protected if the transformation functions used are non-invertible. This is possible if the transformation functions are surjective, non-injective and asymmetric in nature. In our framework, the transformation functions were designed to ensure that they possess the privacy protecting properties. The supporting Lemmas showed that each of the transformations do possess those properties.

The LBSP pushes the advertisements / coupons and sale notifications based on the context of the generalized service request that the LBSP is serving. The query server parses the set of results and also the set of advertisements / coupons / sale notifications relevant to the generic query and / or the centroid locations in the service request. Based on each of the users original request, the query server reconstructs the results and generates personalized results for each user's request. Thus, our framework incorporates the protection of revenue model of the LBSP

by disbursing Context Aware Privacy Protecting Advertisements to the user.

We identified various types of possible attacks and have used a graph theoretic approach to demonstrate that under all possible attack situations, the identity of the user is not revealed to the adversary. Using the Bayesian Chain rule, we have also demonstrated that our framework is better than the existing solutions. Thus we claim that through the use of the novel CLOPRO framework, we have comprehensively addressed the *Identity* privacy, *Location* privacy and *Query* privacy simultaneously while protecting the revenue of the LBSP.

10.1.1 Implementation

For the query anonymization process, we generate generic queries and associate a query code with each of the generic queries. The Qcode generation module is performed off line and a separate Window's application is developed for the Qcode generator. Registration of the user and the LBSP is the first step in the process of using the CLOPRO framework. The message passing is in XML format and so the service requests are simulated through a list of XML messages. The user's request generates an XML message, which is appended to the list of messages that holds the service requests of all the users. Based on the preliminary results, we also created a much more robust and realistic dataset. The algorithms to transform the context and the query of the service request to protect the privacy of the user have been implemented using the simple concepts of clustering and query abstraction through the process of generalization. Since the time required for obtaining the service requests is minimal, it is assumed that the user's location has varied minimally.

10.1.2 Comparison with some existing solutions

We have verified the viability and the scalability of the use of the CLOPRO framework. We have verified the successful operation for over simultaneous requests from 120 user's having a query in a specific area in the small time frame. The limitation is due to the hardware used for implementation. We believe that if the capability of the servers is increased, it might be possible to make the algorithm scalable. As the use of the context aware services increases, the system could be implemented on high end dedicated servers that can be replicated at various locations to improve the performance.The complexity of our *Location Anonymization Algorithm*

(LAA) of *SCAA* is $O(n)$. The complexity of the Cluster Cloak algorithm is $O(nlgn)$ [34] which is lower than Nbr-k, Local-k [7] and ARNN algorithms [64] as mentioned by Yao et. al.

10.2 Future Work

The work done shows that it is possible to implement a framework that protects the privacy of the user comprehensively. However, there are possible extensions to this work as follows:

- ♢ The framework can be modified to incorporate the continuous queries of the user.
- ♢ We have identified one set of transformations that satisfy the property of surjectivity, non-injectivity and asymmetry. More such functions can be identified.
- ♢ We have implemented a simple distance based clustering algorithm. Different set of algorithms can also be implemented.
- ♢ The query abstraction level can be user defined.

REFERENCES

[1] Mobile Marketing Watch by Michael. Available at: "http://www.mobilemarketingwatch.com/google-says-local-intent-is-behind-one-third-of-mobile-searches-5800/"„ Retrieved on 03/08/2013.

[2] Joshua Pramis. Number of mobile phones to exceed world population by 2014, February 28 2013. Retireved on 12/14/2013.

[3] Sergio Mascetti, Dario Freni, Claudio Bettini, X Sean Wang, and Sushil Jajodia. Privacy in geo-social networks: proximity notification with untrusted service providers and curious buddies. *The VLDB journal*, 20(4):541–566, 2011.

[4] Mobile Marketing Watch Google Report. Available at : "http://www.mobilemarketingwatch.com/google-says-local-intent-is-behind-one-third-of-mobile-searches-5800/", Retrieved on March 23 2010.

[5] AlexeiOreskovic. Google removes privacy feature from android mobile software, December 28 2013. Retireved on 12/14/2013.

[6] Whitney Lance CNET News. Available at : "http://news.cnet.com/8301-1023_3-20037170-93.html", Retrieved on 05/09/2013.

[7] Buğra Gedik and Ling Liu. Protecting location privacy with personalized k-anonymity: Architecture and algorithms. *Mobile Computing, IEEE Transactions on*, 7(1):1–18, 2008.

[8] Peter Hornyack, Seungyeop Han, Jaeyeon Jung, Stuart Schechter, and David Wetherall. These aren't the droids you're looking for: retrofitting android to protect data from imperious applications. In *Proceedings of the 18th ACM conference on Computer and communications security*, pages 639–652. ACM, 2011.

[9] Ilias Leontiadis, Christos Efstratiou, Marco Picone, and Cecilia Mascolo. Don't kill my ads!: balancing privacy in an ad-supported mobile application market. In *Proceedings of the Twelfth Workshop on Mobile Computing Systems & Applications*, page 2. ACM, 2012.

[10] RssPhone.com_Smartphone news "http://www.rssphone.com/google-play-store-800000-apps-and-overtake-apple appstore/". Retrieved on 05/09/2013.

[11] Saikat Guha, Mudit Jain, and Venkata Padmanabhan. Koi: a location-privacy platform for smartphone apps. In *Proceedings of the 9th Symposium on Networked Systems Design and Implementation (NSDI)*, 2012.

[12] Chowdhury S Hasan, Sheikh I Ahamed, and Mohammad Tanviruzzaman. A privacy enhancing approach for identity inference protection in location-based services. In *Computer Software and Applications Conference, 2009. COMPSAC'09. 33rd Annual IEEE International*, volume 1, pages 1–10. IEEE, 2009.

[13] Latanya Sweeney. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(05):557–570, 2002.

[14] Panos Kalnis, Gabriel Ghinita, Kyriakos Mouratidis, and Dimitris Papadias. Preventing location-based identity inference in anonymous spatial queries. *Knowledge and Data Engineering, IEEE Transactions on*, 19(12):1719–1733, 2007.

[15] AA Pandit and Anup Kumar. Conceptual framework and a critical review for privacy preservation in context aware systems. In *Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), 2012 International Conference on*, pages 435–442. IEEE, 2012.

[16] Amir Salar Amoli, Mehdi Kharrazi, and Rasool Jalili. 2ploc: Preserving privacy in location-based services. In *Social Computing (SocialCom), 2010 IEEE Second International Conference on*, pages 707–712. IEEE, 2010.

[17] Man Lung Yiu, Christian S Jensen, Xuegang Huang, and Hua Lu. Spacetwist: Managing the trade-offs among location privacy, query performance, and query accuracy in mobile services. In *Data Engineering, 2008. ICDE 2008. IEEE 24th International Conference on*, pages 366–375. IEEE, 2008.

[18] Toby Xu and Ying Cai. Location cloaking for safety protection of ad hoc networks. In *INFOCOM 2009, IEEE*, pages 1944–1952. IEEE, 2009.

[19] Marco Gruteser and Dirk Grunwald. Anonymous usage of location-based services through spatial and temporal cloaking. In *Proceedings of the 1st international conference on Mobile systems, applications and services*, pages 31–42. ACM, 2003.

[20] Claudio A Ardagna, Marco Cremonini, Sabrina De Capitani di Vimercati, and Pierangela Samarati. An obfuscation-based approach for protecting location privacy. *Dependable and Secure Computing, IEEE Transactions on*, 8(1):13–27, 2011.

[21] Gabriel Ghinita, Panos Kalnis, Ali Khoshgozaran, Cyrus Shahabi, and Kian-Lee Tan. Private queries in location based services: anonymizers are not necessary. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 121–132. ACM, 2008.

[22] Michael J Freedman, Emil Sit, Josh Cates, and Robert Morris. Introducing tarzan, a peer-to-peer anonymizing network layer. In *Peer-to-Peer Systems*, pages 121–129. Springer, 2002.

[23] David Goldschlag, Michael Reed, and Paul Syverson. Onion routing. *Communications of the ACM*, 42(2):39–41, 1999.

[24] Vijayalakshmi Atluri and Heechang Shin. Efficient security policy enforcement in a location based service environment. In *Data and Applications Security XXI*, pages 61–76. Springer, 2007.

[25] Arun Kumar, Neeran Karnik, and Girish Chafle. Context sensitivity in role-based access control. *ACM SIGOPS Operating Systems Review*, 36(3):53–66, 2002.

[26] Chiel Drost. Privacy in context-aware systems. *University of Twente, Enschede, Netherlands, Department of Informatics, Federal University of Espirito Santo, Vitoria, Brazil. Retrieved November*, 5:2011, 2004.

[27] A Corrad, Rebecca Montanari, and Daniela Tibaldi. Context-based access control management in ubiquitous environments. In *Network Computing and Applications, 2004.(NCA 2004). Proceedings. Third IEEE International Symposium on*, pages 253–260. IEEE, 2004.

[28] John Andrew Simpson, Edmund SC Weiner, et al. *The Oxford english dictionary*, volume 2. Clarendon Press Oxford, 1989.

[29] Andreas Pfitzmann and Marit Köhntopp. Anonymity, unobservability, and pseudonymity: a proposal for terminology. In *Designing privacy enhancing technologies*, pages 1–9. Springer, 2001.

[30] Ashwin Machanavajjhala, Daniel Kifer, Johannes Gehrke, and Muthuramakrishnan Venkitasubramaniam. l-diversity: Privacy beyond k-anonymity. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 1(1):3, 2007.

[31] Xiaokui Xiao and Yufei Tao. M-invariance: towards privacy preserving re-publication of dynamic datasets. In *Proceedings of the 2007 ACM SIGMOD international conference on Management of data*, pages 689–700. ACM, 2007.

[32] Amirreza Masoumzadeh and James Joshi. An alternative approach to< i> k</i>-anonymity for location-based services. *Procedia Computer Science*, 5:522–530, 2011.

[33] Mohamed F Mokbel, Chi-Yin Chow, and Walid G Aref. The new casper: query processing for location services without compromising privacy. In *Proceedings of the 32nd international conference on Very large data bases*, pages 763–774. VLDB Endowment, 2006.

[34] Lin Yao, Chi Lin, Xiangwei Kong, Feng Xia, and Guowei Wu. A clustering-based location privacy protection scheme for pervasive computing. In *Proceedings of the 2010 IEEE/ACM Int'l Conference on Green Computing and Communications & Int'l Conference on Cyber, Physical and Social Computing*, pages 719–726. IEEE Computer Society, 2010.

[35] Hidetoshi Kido, Yutaka Yanagisawa, and Tetsuji Satoh. An anonymous communication technique using dummies for location-based services. In *Pervasive Services, 2005. ICPS'05. Proceedings. International Conference on*, pages 88–97. IEEE, 2005.

[36] Hua Lu, Christian S Jensen, and Man Lung Yiu. Pad: Privacy-area aware, dummy-based location privacy in mobile services. In *Proceedings of the Seventh ACM International Workshop on Data Engineering for Wireless and Mobile Access*, pages 16–23. ACM, 2008.

[37] Gabriel Ghinita, Panos Kalnis, and Spiros Skiadopoulos. Mobihide: a mobilea peer-to-peer system for anonymous location-based queries. In *Advances in Spatial and Temporal Databases*, pages 221–238. Springer, 2007.

[38] Tai Cheng Li and Wen Tao Zhu. Protecting user anonymity in location-based services with fragmented cloaking region. In *Computer Science and Automation Engineering (CSAE), 2012 IEEE International Conference on*, volume 3, pages 227–231. IEEE, 2012.

[39] Tinghuai Ma, Sen Yang, Wei Tian, and Wenjie Liu. Privacy preserving in ubiquitous computing: Architecture. *Information Technology Journal*, 8(6):910–916, 2009.

[40] Toby Xu and Ying Cai. Exploring historical location data for anonymity preservation in location-based services. In *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*, pages 547–555. IEEE, 2008.

[41] Masanori Mano and Yoshiharu Ishikawa. Anonymizing user location and profile information for privacy-aware mobile services. In *Proceedings of the 2nd ACM SIGSPATIAL International Workshop on Location Based Social Networks*, pages 68–75. ACM, 2010.

[42] Chi-Yin Chow, Mohamed F Mokbel, and Xuan Liu. A peer-to-peer spatial cloaking algorithm for anonymous location-based service. In *Proceedings of the 14th annual ACM international symposium on Advances in geographic information systems*, pages 171–178. ACM, 2006.

[43] Chi Lin, Guowei Wu, Lin Yao, and Zuosong Liu. A combined clustering scheme for protecting location privacy and query privacy in pervasive environments. In *Trust, Security and Privacy in Computing and Communications (TrustCom), 2012 IEEE 11th International Conference on*, pages 943–948. IEEE, 2012.

[44] Piotr Indyk and David Woodruff. Polylogarithmic private approximations and efficient matching. In *Theory of Cryptography*, pages 245–264. Springer, 2006.

[45] Ali Khoshgozaran and Cyrus Shahabi. Blind evaluation of nearest neighbor queries using space transformation to preserve location privacy. In *Advances in Spatial and Temporal Databases*, pages 239–257. Springer, 2007.

[46] Matt Duckham and Lars Kulik. A formal model of obfuscation and negotiation for location privacy. In *Pervasive Computing*, pages 152–170. Springer, 2005.

[47] Matt Duckham and Lars Kulik. Simulation of obfuscation and negotiation for location privacy. In *Spatial Information Theory*, pages 31–48. Springer, 2005.

[48] Claudio Agostino Ardagna, Marco Cremonini, Ernesto Damiani, S De Capitani di Vimercati, and Pierangela Samarati. Location privacy protection through obfuscation-based techniques. In *Data and Applications Security XXI*, pages 47–60. Springer, 2007.

[49] Ryan Wishart, Karen Henricksen, and Jadwiga Indulska. Context privacy and obfuscation supported by dynamic context source discovery and processing in a context management system. In *Ubiquitous Intelligence and Computing*, pages 929–940. Springer, 2007.

[50] Gabriel Ghinita, Panos Kalnis, and Spiros Skiadopoulos. Prive: anonymous location-based queries in distributed mobile systems. In *Proceedings of the 16th international conference on World Wide Web*, pages 371–380. ACM, 2007.

[51] Aniket Pingley, Wei Yu, Nan Zhang, Xinwen Fu, and Wei Zhao. Cap: A context-aware privacy protection system for location-based services. In *Distributed Computing Systems, 2009. ICDCS'09. 29th IEEE International Conference on*, pages 49–57. IEEE, 2009.

[52] Jalal Al-Muhtadi, Roy Campbell, Apu Kapadia, M Dennis Mickunas, and Seung Yi. Routing through the mist: Privacy preserving communication in ubiquitous computing environments. In *Distributed Computing Systems, 2002. Proceedings. 22nd International Conference on*, pages 74–83. IEEE, 2002.

[53] Daniele Riboni, Linda Pareschi, and Claudio Bettini. Privacy in georeferenced context-aware services: A survey. In *Privacy in Location-Based Applications*, pages 151–172. Springer, 2009.

[54] Urs Hengartner and Peter Steenkiste. Implementing access control to people location information. In *Proceedings of the ninth ACM symposium on Access control models and technologies*, pages 11–20. ACM, 2004.

[55] Richard Hull, Bharat Kumar, Daniel Lieuwen, Peter F Patel-Schneider, Arnaud Sahuguet, Sriram Varadarajan, and Avinash Vyas. Enabling context-aware and privacy-conscious user data sharing. In *Mobile Data Management, 2004. Proceedings. 2004 IEEE International Conference on*, pages 187–198. IEEE, 2004.

[56] Richard Cissée Andreas Rieger Nicolas Braun and Sahin Albayrak. An agent-based framework for secure and privacy-preserving personalized information services. *Revista Colombiana de Computación (RCC)*, 7(1), 2006.

[57] Pramod Jagtap, Anupam Joshi, Tim Finin, and Laura Zavala. Preserving privacy in context-aware systems. In *Semantic Computing (ICSC), 2011 Fifth IEEE International Conference on*, pages 149–153. IEEE, 2011.

[58] Jason I Hong. The context fabric: an infrastructure for context-aware computing. In *CHI'02 Extended Abstracts on Human Factors in Computing Systems*, pages 554–555. ACM, 2002.

[59] Yu Chen, Jie Bao, Wei-Shinn Ku, and Jiun-Long Huang. Cache management techniques for privacy preserving location-based services. In *Mobile Data Management Workshops, 2008. MDMW 2008. Ninth International Conference on*, pages 88–96. IEEE, 2008.

[60] Alastair R Beresford, Andrew Rice, Nicholas Skehin, and Ripduman Sohan. Mockdroid: trading privacy for application functionality on smartphones. In *Proceedings of the 12th Workshop on Mobile Computing Systems and Applications*, pages 49–54. ACM, 2011.

[61] Mohammad Nauman, Sohail Khan, and Xinwen Zhang. Apex: extending android permission model and enforcement with user-defined runtime constraints. In *Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security*, pages 328–332. ACM, 2010.

[62] Mehdi Behzad, 1936 Chartrand, Gary, and Linda Lesniak-Foster. *Graphs & digraphs / Mehdi Behzad, Gary Chartrand, Linda Lesniak-Foster*. Boston : Prindle, Weber & Schmidt, 1979. Bibliography: p. [383]-401.

[63] Robert Tarjan. Depth-first search and linear graph algorithms. *SIAM journal on computing*, 1(2):146–160, 1972.

[64] Chi-Yin Chow, Mohamed F Mokbel, Joe Naps, and Suman Nath. Approximate evaluation of range nearest neighbor queries with quality guarantee. In *Advances in Spatial and Temporal Databases*, pages 283–301. Springer, 2009.

CURRICULUM VITAE

NAME	ANALA ANIRUDDHA PANDIT
E-MAIL	adkulk01@louisville.edu

EDUCATION

- ♢ M.Sc. (Physics with Specialization in Electronics) University of Mumbai, 1983
- ♢ M.S. (Electrical Engg) University of Louisville, 1985
- ♢ M.S., (Computer Science) University of Louisville, 2012

WORK EXPERIENCE

Research Assistant
University of Louisville, MINDS Lab (RTDSS Project)
July 2011 - September 2013

AWARDS, MEMBERSHIPS AND SERVICE

- ♢ IEEE Outstanding Student Award 2011
- ♢ IEEE Student Member
- ♢ AIRC Vice President - August 2011 - August 2013
- ♢ SIGS Ambassador - Univ. of Louisville - August 2012 - till date
- ♢ GSC Representative (CECS) - August 2013 - till date

Selected Publications and Research

- Pandit, A. A., and Kantardzic M. M. "New hierarchical model using SOM improves accuracy of classifiers for multiclass data sets", Information, Communication and Automation Technologies (ICAT), 2011 XXIII International Symposium on. IEEE, 2011.
- Pandit, A. A., and Kumar. A. "Interactive Context-Aware Services for Mobile Devices", Handbook of Mobile Systems Applications and Services 1 (2012): 91.
- Pandit, A. A., and Kumar. A. "Conceptual Framework and a Critical Review for Privacy Preservation in Context Aware Systems", Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), 2012 International Conference on. IEEE, 2012.
- Pandit A. A., and Kantardzic M. M. “HIERSOMOVA: A New Hierarchical Approach in Multiclass Classification.” Poster, SAS Data Analytics 2013.
- Pandit, A. A.; Nutakki, G .C, and Nasraoui, O. “Privacy Signal: A Data-Driven Browser Extension for Online Privacy Policy Safety Indication”, Transactions on Information Forensics & Security, Under Peer review
- Pandit, A. A., Polina, P., Kumar A. “CLOPRO: A framework for Context Cloaking Privacy Protection”, Fourth International Conference on Communication Systems and Network Technologies 2014, Accepted.
- Pandit A. A., Polina P., Kumar A., Xie. B., “CAPPA: Context Aware Privacy Protecting Advertising Extension to CLOPRO Framework”, 11th IEEE International Conference on Services Computing June 27 - July 2, 2014, Anchorage, Alaska, USA, Submitted.

PROJECTS "Privacy Signal: A Browser Extension for Online Privacy Policy Safety Indication" - Partial Fulfillment for M.S. (CS) University of Louisville.