# PRIVACY AWARE COLLABORATIVE TRAFFIC MONITORING VIA ANONYMOUS ACCESS AND AUTONOMOUS LOCATION UPDATE MECHANISM

by

Belal Mohammed Amro

Submitted to the Graduate School of Engineering and Natural Sciences

in partial fulfillment of

the requirements for the degree of

Doctor of Philosophy

Sabanci University

August 2012

# PRIVACY AWARE COLLABORATIVE TRAFFIC MONITORING VIA ANONYMOUS ACCESS AND AUTONOMOUS LOCATION UPDATE MECHANISM

APPROVED BY:

Assoc. Prof. Dr. Yücel Saygın ……………………………………

(Dissertation Supervisor)

Assoc. Prof. Dr. Albert Levi ……………………………………

(Dissertation Co-supervisor)

Assist. Prof. Dr. Ali İnan ……………………………………

Assist. Prof. Dr. Ercan Nergiz ……………………………………

Assoc. Prof. Dr. Özgür Gürbüz ……………………………………

Date of approval 2/8 / 2012

# ABSTRACT

Collaborative Traffic Monitoring, CTM, systems collect information from users in the aim of generating a global picture of traffic status. Users send their location information including speed and directions, and in return they get reports about traffic in certain regions. There are two major approaches for the deployment of CTM systems. The first approach relies on dedicated communication infrastructure (DI). This approach is still being investigated by researchers and there is no important deployments done yet. The other approach utilizes existing communication infrastructures (EI) such as Wi-Fi, GSM, and GPRS for communication between users and traffic server.

Due to the sensitivity of location information, different privacy preserving techniques have been proposed for both DI and EI approaches. In DI approach the concentration was on anonymous access using pseudonyms. In EI approach privacy techniques concentrate on hiding the identity of a particular user within other *k*-1 users at the same region or time stamp by using *cloaking*. Cloaking means generalization of location or time stamp so that other *k*-1 users will have the same generalized value. Unfortunately, cloaking decreases the quality of the data and requires a Trusted Third Party (TTP) to determine the cloaked region or cloaked time stamp.

In this thesis, we propose a Privacy Aware Collaborative Traffic Monitoring System (PA-CTM) that considers the privacy and security properties of VANETs and existing infrastructures. PA-CTM provides a client server architecture that relies on existing infrastructures and enhances privacy by (1) Using a robust Collusion Resistant Pseudonym Providing System, CoRPPS, for anonymous access. Users are able to change their pseudonyms and hence hide their complete trajectory information form traffic server; (2) Utilizing a novel Autonomous Location Update Mechanism, ALUM, that does not rely on a Trusted Third Party and uses only local parameters (speed and direction) for triggering a location update or pseudonym change. Our performance results showed that CoRPPS provides a high level of anonymity with strong resistant against collusion attacks. Performance results also showed that ALUM is effective for traffic monitoring in terms of both privacy and utility.

# ÖZET

İşbirlikçi Trafik İzleme, İTİ, sistemleri trafik durumunun geniş çaplı resmini oluşturmak amacıyla kullanıcılardan bilgi toplarlar. Kullanıcılardan gelen hız ve yönleriyle beraber konum bilgilerini yorumlayan bu sistemler, karşılık olarak istenilen bölgelerdeki trafik durumu hakkında rapor gönderirler. İTİ sistemlerinin konuşlandırılması için iki temel yaklaşım vardır. İlk yaklaşım özel iletişim altyapısı'na (ÖA) dayanır. Araştırmacılar tarafından halen incelenmekte olan bu yaklaşımın henüz önemli bir konuşlandırması bulunmamaktadır. Diğer yaklaşım ise kullanıcılar ve trafik sunucusu arasındaki iletişim için Wi-Fi, GSM ve GPRS gibi mevcut iletişim altyapılarını (MA) kullanır.

Konum bilgisinin hassasiyeti nedeniyle, ÖA ve MA yaklaşımlarının her ikisi için de farklı mahremiyet koruma teknikleri önerilmiştir. ÖA yaklaşımında mahlas kullanarak anonim erişim sağlamaya önem verilmiştir. MA yaklaşımında ise aynı alan veya zaman damgası içerisindeki *k* farklı kullanıcı arasından belirli bir kullanıcının kimliğini saklamak için geri kalan *k*-1 kullanıcı *perdeleme* görevi görür. Yer veya zaman damgası bilgisinin genelleştirilmesini sağlayan perdeleme yöntemi sayesinde, geriye kalan *k*-1 kullanıcı aynı genelleştirilmiş değerlere sahip olmaktadır. Ne yazık ki, perdeleme yöntemi verilerin kalitesini düşürmekte ve perdelenmiş yer veya zaman damgası bilgisinin belirlenmesi için Güvenilir Üçüncü Parti'ye (GÜP) ihtiyaç duymaktadır.

Bu tezde, VANET'lerin ve mevcut altyapıların mahremiyet ve güvenlik özelliklerini gözeten bir Mahremiyet Bilinçli İşbirlikçi Trafik İzleme (MB-İTİ) sistemi öneriyoruz. MB-İTİ mevcut altyapılara dayanan bir istemci sunucu mimarisi ile mahremiyeti artırmak için (1) anonim erişim için güçlü bir Danışıklı Hileye Dayanıklı Mahlas Sağlama Sistemi, DHDMSS, kullanır. Kullanıcılar mahlaslarını değiştirebildiklerinden dolayı izledikleri yörüngeyi trafik sunucularından saklayabilirler; (2) Güvenilir Üçüncü Parti'ye ihtiyaç duymayan ve konum güncelleme veya mahlas değişikliği için sadece yerel parametrelerden (hız ve yön) faydalanan orjinal bir Özerk Yer Güncelleme Mekanizması (ÖYGM) kullanır. Performans sonuçlarımız, DHDMSS yüksek düzeyde anonimlik ile beraber danışıklı hile saldırılarına karşı güçlü bir direnç sağladığını göstermiştir. Aynı zamanda, performans sonuçlarımız ÖYGM’nın mahremiyet ve hizmet bakımından trafik izleme için etkili olduğunu da göstermiştir.

## DEDICATION

To my parents, My wife SAMAR , and my children WADEE and BASHEER

## ACKNOWLEDGEMENTS

I am glad to be able to thank everybody who supported me during my PhD study since February 2008. Firstly, and with all proud, I would like to thank my primary supervisor Prof. Yücel Saygın and my co-supervisor Prof. Albert Levi for their tenacious support during my research development and dissertation preparation. I deeply appreciate their guidance, encouragement, and support which made my dissertation possible.

I would like to express my sincere thanks to Prof. Özgür Gürbüz and Prof. Ali Inan for being my thesis proposal committee and for their valuable comments and suggestions during my proposal defense. I would like also to thank Prof Ercan Nergiz for his contribution in my thesis committee.

I gratefully acknowledge the funding sources that made my Ph.D. work possible. I was funded by the ERASMUS MUNDUS project, which is sponsored by the European Union, for my first 34 months. I also was honored to get a full teacher assistant scholarship at Sabanci University for the rest of my study.

My time at Sabanci University was made enjoyable in large part due to the many friends who became a part of my life. My deep acknowledgments are for Yassir El-Kahlout, Ahmed Adbalaal, and Iyad Hashlamon for their support and encouragement. We were like a family and spent most time together. They were the source of encouragement and happiness for me at Sabanci University especially during my sickness period.

I should not forget to thank my colleagues at Hebron University for their support and encouragements. Special thanks for Prof. Salman Talahmeh and Prof. Nabil Hasasneh for their invaluable support and encouragement.

Lastly, I would like to thank my family for all their love and encouragement. For my parents who raised me with love and supported me in all my pursuits. And most of all for my loving, supportive, encouraging, and patient wife SAMAR whose faithful support during my study of this Ph.D. is so appreciated. Thank you SAMAR for your patience during my alienation and for raising our children Wadee and Basheer during my absence. I

feel very sorry for being away from my wife and children during my study, and I want to apologize for them for this and hope that I can compensate them those dark days.

Finally, I would like to dedicate the following quote to all people who have helped me personally and professionally to pursue my PhD degree.

"*At times our own light goes out and is rekindled by a spark from another person. Each of us has cause to think with deep gratitude of those who have lighted the flame within us"- Albert Schweitzer*

# TABLE OF CONTENTS

## LIST OF FIGURES

## LIST OF TABLES

## SYMBOLS AND ABBREVIATIONS

| *Notation* | **Meaning** |
|---|---|
| $PS$ | Pseudonym Signer |
| $SP$ | Service Provider |
| $AS$ | Authentication Server |
| $RA$ | Registration Authority |
| $ID_U$ | Identity of generic user (assigned by $RA$ randomly and do not carry any information about the real identity) |
| $ID_{AS}$ | Identity of generic $AS$ |
| $AS_i$ | $AS$ whose $ID_{AS} = i$ |
| $GID$ | Identity of Generic Group |
| $GID_U$ | The identity of a particular user's group. |
| $\mathbb{G}_i$ | Set of $AS$s belonging to a group with $GID = i$. |
| $Ctr_U$ | The counter value of a particular user $U$ |
| $Ctr_{max}$ | The upper limit of users' counters |
| $K$ | Symmetric Encryption key shared between $AS$s and $PS$ |
| $t$ | token from token pool |

| | |
|---|---|
| $v$ | Verification code in the range of [0, $V_{max}$) |
| $E_K(t\|\|v)$ | $t$ concatenated with $v$ and encrypted with $K$, also called a *ticket* |
| $\mathbb{T}$ | Set of tokens extracted from a particular user's tickets issued by $AS$s at a particular $Ctr_U$ value |
| $pw_U^{AS}$ | password shared between of user $U$ and a particular $AS$. This value is generated by $RA$ during registration |
| $t_{comb}$ | The combination of $(\mathbb{T}, v)$ |
| $N_{t_{comb}}$ | Total number of possible $t_{comb}$ values |
| $\mathbb{TC}_U$ | Set of all $t_{comb}$s issued by $AS$s of the user $U$'s group |
| $\mathbb{RP}_U$ | Set of all pseudonyms signed by $PS$ for the user $U$. |
| $P_U^i$ | $i^{th}$ pseudonym of user $U$ |
| $\sigma_{SK_{PS}}(P_U^i)$ | The signature of $PS$ over $P_U^i$ |
| $g$ | Number of $AS$s in a group |
| $N_{AS}$ | Total number of $AS$s |
| $N_U$ | Total number of users |
| $N_t$ | Total number of tokens in token pool |
| $hash(\cdot)$ | Secure hash function |
| $NP_{max}$ | Maximum number of pseudonyms signed per each trial |

$VT_{period}$ Time period at which pseudonyms of a particular trial are valid through

$VF_{period}$ Time period at which a particular pseudonym is valid from the first t time it is used

$TTP$ Trusted Third Party

$A.updatetime$ The time moving object *A* has updated her location

$A.value$ The location of moving object *A* at time *A.updatetime*

$A.function$ The speed of moving object *A* at time *A.updatetime*

$A.direction$ The direction of moving object *A* at time *A.updatetime*

$SD_{Thresh}$ The threshold value of speed change. It helps triggering a location update and a pseudonym change.

$W_{Thresh}$ The threshold value of the weight of a particular sub region, it describes the traffic activity in that sub region

$SF$ San Francisco city

$UR$ Uncertainties Region, region where all vehicles have the same probability of having updated their locations.

$GPS_{ERR}$ GPS Error, Error due to imprecision of GPS receivers.

$Inh_{ERR}$ Inherent error, occurs due to errors from previous location calculations.

$LocPrec_{ERR}$ Location Precision Error, Error due to calculations of expected location of a particular vehicle.

| | |
|---|---|
| $FTL$ | Future Temporal Language, a query language based on future temporal logic. |
| $RAC$ | Relative Area Coverage |
| $WRC$ | Weighted Area Coverage |
| $RCC$ | Relative Communication Cost |
| $MAX_U$ | Maximum number of users |
| $L_{RC}$ | The length of random challenge in bits |
| $L_{UR}$ | The length of user response in bits |
| $L_t$ | The length a ticket in bits |
| $L_P$ | The length a pseudonym in bits |
| $TA_L$ | Traffic size for authentication and tickets acquisition |
| $PS_L$ | Traffic size for singing a bunch of pseudonyms |

# 1. INTRODUCTION

Traffic monitoring systems have evolved rapidly in the last years due to the advances in communication technologies such as GPS, GSM and 3G networks. The main idea behind Collaborative Traffic Monitoring (CTM) systems is that users provide their location information to have a global model of the current traffic [1]. CTM systems are critical nowadays especially in big cities with heavy and sometimes unpredictable traffic. However, privacy is considered a major obstacle in front of turnouts of users to these systems [2,3].

## 1.1 Motivation

In this Section, we first list the driving forces behind the widespread of CTM systems. Then we list the motivation towards deployment of a privacy preserving CTM system.

### 1.1.1 Motivation for Collaborative Traffic Monitoring

Collaborative Traffic Monitoring (CTM) has recently become a hot research topic for the great benefits such as time and energy saving, environmental protection, and traffic safety. The main driving force of CTM is the rapid increase of the amount of vehicles relative to new road openings [4]. CTM systems utilize disseminated information to save time for system users by providing them with route information and expected delays. Besides the time savings, it also saves fuel consumption by decreasing the waiting time while engine is on. Royal Automobile Club of Queensland in Australia (RACQ) reported that fuel consumption increases by 30% when there is congestion in traffic [6]. The Parliamentary Office of Science and Technology [5], has reported that about 44% of congestion may be avoided using CTM systems. It has also reported significant results about fatal accident reductions and money savings as well.

CTM systems also help decrease pollution and carbon monoxide (CO) and carbon dioxide ($CO_2$) levels due to less waiting time on traffic queues. As a remedy for air pollution in Southern California, the Association of Governments suggested improving transportation system by utilizing CTM systems [6].

Many accidents can be avoided by providing emergency messages for vehicles in the neighborhood. This implies saving lives and money. According to CARE reports, it was found that 60% of accidents are caused by driver behavior [7]. This means that these accidents can be reduced by providing drivers with useful and emergency information.

Many insurance companies have provided new policies regarding the driving behavior of policy holders. Insurance companies may decrease the policy cost of the driver according to her driving behavior. These insurance companies can rely on CTM systems for generating the driver behavior [8].

### 1.1.2 Motivation for Privacy Preserving CTM Systems

Privacy is defined as “the ability of an individual or group to seclude themselves or information about themselves and thereby revealing themselves selectively”[1].

Location privacy is defined as “The ability of an individual to move in public space with the expectation that under normal circumstances her location will not be systematically and secretly recorded for later use” [9].

People do not want being virtually tracked while they are driving so that no one can identify them using their routes.  Therefore, their movement information should be hidden from others. Otherwise, privacy requirements of CTM users cannot be fulfilled.

In his very informative lecture about location privacy in mobile world, Al Gidari [10] gave plenty of examples on how location information can be used to reveal lots of private information. He also recommended changing  the law that governs  location information

[1] http://en.wikipedia.org/wiki/Personal_privacy

history in United States of America with a new standard that addresses all possible directions such as the duration of storing data, how frequently to answer the query, etc.

CTM systems require the users to provide their exact locations periodically to come up with an accurate traffic estimate. This location disclosure may reveal lots of private information of CTM users including route disclosure of a particular user. It also enables user profiling by gathering information of places of interest for that user [2,11].

The widespread use of smart phones with GPS technologies made the tracking of users easier by providing their exact locations together with their timestamps. This crowded data carry huge risks of privacy leakage, which should be considered in designing CTM systems. However, the existence of these mobile phone networks reduced the infrastructure cost required for building the CTM systems by utilizing existing networks rather than establishing new dedicated ones [12].

Also Patrick [13] did a study on the concept of Ambient Intelligence (AmI). The concept AmI arises from the convergence of ubiquitous computing, ubiquitous communication, and intelligent user friendly interface. This implies that a person is surrounded by computing and networking technologies that are aware of his presence. He analyzed the concept "AmI" over European Union data protection law. Then he used his analysis to develop an argument for all regulatory solutions that enforce protection of private data. The paper concluded that "AmI" concept presents a significant threat to personal privacy.

An interesting study was done by Cvrcek et al. [14] on some European countries about the price of their location privacy for different periods of time. The study showed that good percentage of people are aware of their location privacy and deal carefully regarding that issue.

In their website[2], Electronic Frontier Foundation (EFF) published an essay regarding location privacy. They said that it is not only the government that people have to be afraid

[2] www.eff.org

of disclosing location information to, but also they need to hide information from other people. They gave different examples about that. They also concluded that this is the time for organizations to show leadership and select designs that respects and protects users' privacy [9].

From the above, we have no doubt about the importance of privacy in the presence of data mining tools. The privacy risk involves different parts of society starting from regular people up to companies and even political parties. Few examples include having a girl friend while being married. Also political communications between parties may be disclosed too. Companies' communications may be revealed by tracking CEO's and their meetings; this may affect the shares of involved companies. One can imagine different scenarios for different parts of society which at the end lead to affect the whole society.

For a more concrete example, consider the following scenario. Mr. X is a teacher working in a school somewhere in a city. He used to go from his house to school and return back regularly. Recently, he started to visit a cancer medical center regularly and stays there for hours. Mr. X was planning to buy a life insurance policy before he was diagnosed. If the insurance agents infer his periodic visits to the cancer center, such breach of location info may affect the price of his insurance policy or even the refusal of selling him the policy. This also leads to a disclosure of being infected with cancer. Of course Mr. X does not want anyone to know about his disease. This example is one of many scenarios that include privacy violation using location information.

## 1.2 Objectives of the Thesis

Existing CTM solutions generally use two different methodologies. The first one is the dedicated infrastructure approach, also called VANETs (Vehicular Ad Hoc Networks), where a dedicated infrastructure for communication is deployed; we call this approach DI for short. The second methodology utilizes existing wireless networks, such as GSM, GPRS, EDGE, UMTS and Wi-Fi; we call this approach Existence Infrastructure, EI. DI requires investments in deployments of the dedicated infrastructure that is not widely done yet.

DI users use pseudonyms for anonymous access to traffic server. DI approaches concentrate on anonymous access for preserving privacy and do not concentrate well on preserving privacy of location information [21,29,30,31].

On the other hand, EI approaches utilize different mechanisms for preserving location information privacy with little concentration on anonymous access. Our objective is to develop an EI CTM system that is equivalent (in terms of privacy and security) to the DI approach, i.e. a CTM system that combines both anonymous access of DI and location information privacy mechanisms of EI. The challenge is to design a system that allows anonymous access for users and maintains a back door for identity revealing under law enforcement purposes only. Another challenge is to protect anonymous users from being identified via their location information. Overall, the system should be efficient for traffic monitoring in terms of utility metrics.

## 1.3 Contributions

The aim of the thesis is to build a Privacy Aware Collaborative Traffic Monitoring System (PA-CTM), that is aware of users' privacy and depends on existing communication infrastructures instead of having dedicated infrastructure. The design of our PA-CTM is divided into two stages: (1) The first stage is the design of a Collusion Resistant Pseudonym Providing System (CoRPPS). CoRPPS will be used to register users and provide them with pseudonyms that enable them to anonymously access traffic server. (2) The second stage is the design of a novel Autonomous Location Update Mechanism (ALUM) that enhances privacy without depending on Trusted Third Parties (TTPs). ALUM controls the location update and pseudonym change to enhance the privacy level of users and avoid privacy leakage using spatiotemporal data.

In this first contribution, we have designed a novel collusion resistant anonymous access system called CoRPPS [15,16]. CoRPPS enables users to anonymously access a service while maintaining a backdoor for identity revealing under law enforcement purposes only. Identity revealing in CoRPPS is fair, i.e. it is neither punitive in a way that it allows TTPs to reveal past and future anonymity of a particular user, nor restrictive in a way that it allows revealing only current pseudonym. CoRPPS distributes trust among different entities and maintains a level of anonymity for users. Collusion among a subset of these entities, in the aim of revealing a real identity, is avoided in CoRPPS. The backdoor of identity revealing for law enforcement purposes works only when all of the trusted entities participate in the process. CoRPPS is also flexible and can be applied to different anonymous access services by tuning CoRPPS parameters accordingly. Experimental results show that CoRPPS is resistant to collusions among its trusted parties. CoRPPS guarantees a level of anonymity for users at each authentication server. CoRPPS will be used as the pseudonym providing system for our Privacy Aware Collaborative Traffic Monitoring system.

In the second contribution, we developed an Autonomous Location Update Mechanism, ALUM, which enhances location privacy for users without the need for a TTP. ALUM relies only on local parameters (speed and direction) for triggering a location update and a pseudonym change and does not need to communicate with other parties [17,18]. By utilizing local parameters, ALUM is able to avoid redundant location updates and hence reduce communication cost which is a major factor in the widespread of CTM system. Experimental results show that ALUM enhances privacy while maintaining a good level of area coverage and reducing communication cost.

## 1.4 Structure of the Thesis

Chapter 2 introduces background about CTM. It describes properties of both Dedicated and Existing communication Infrastructure CTM systems (DI and EI respectively). Different location update mechanisms are reported with their pros and cons. Related works for both EI and DI approaches are reported as well. We also provid a general description of our privacy preserving CTM design. An introduction to our privacy aware CTM system is presented in Chapter 3. In Chapter 4 we present our Collusion Resistant Pseudonym Providing System, CoRPPS. We introduce the design, communication and flow, properties, and resistant against attacks. Chapter 5 reports the performance analysis of CoRPPS. Analysis includes anonymity, collision probability, and collusion among authentication servers. Autonomous Location Update Mechanism, ALUM, is provided in Chapter 6. The main idea of ALUM is introduced. We also introduce an enhanced mechanism called EALUM. Experimental results for ALUM are provided in Chapter 7. These analyses include $k$-anonymity, Relative Area Coverage, $RAC$, and Relative Communication Cost, $RCC$. Finally, Chapter 8 concludes the work and highlights future research directions.

## 1.5 Summary

In this chapter, we gave an introduction of our thesis, the objectives and motivations as well as expected contributions of our thesis. The structure of our thesis is also provided. In the next chapter we will provide a background and an intensive survey of related work. We will also list the different approaches used in Collaborative traffic monitoring systems, as well as describing different location update mechanisms used in these systems.

# 2. BACKGROUND

In this chapter, we introduce background of Collaborative Traffic Monitoring, CTM, systems basics. The background includes CTM systems communications infrastructures and their properties, recent deployments of CTM systems, and location update mechanisms and privacy issues of CTM systems.

## 2.1 Collaborative Traffic Monitoring Systems

The main idea behind Collaborative Traffic Monitoring (CTM) systems is that users provide their location information to obtain a global view of the current status of traffic. CTM systems are critical nowadays especially in big cities with heavy and sometimes unpredictable traffic. Widespread usage of CTM systems would alleviate the congestion in big cities by proposing alternative routes to the users and avoiding more cars entering the congested areas. In this way, CTM systems would save time and money, and more importantly decrease carbon emission by optimizing the traffic. CTM systems depend on a basic architecture that specifies how entities communicate together

## 2.2 Communication Infrastructures of CTM Systems

CTM systems use client-server architecture. Clients send their location information to a traffic server; the latter provides clients with a real time map about traffic in vicinity [1]. There are two main communication infrastructure approaches for CTM systems. The first one is the Dedicated Infrastructure (DI) approach, this approach is also called VANETs (Vehicular Ad Hoc Networks), where a dedicated infrastructure for communication is deployed [2,19,20,21,22]. The second methodology utilizes existing wireless networks, such as GSM, GPRS, EDGE, UMTS and Wi-Fi for communication with traffic server [17,23,24,25].

### 2.2.1 Vehicular Ad-Hoc Network Dedicated Communication Infrastructure

The Vehicular Ad-Hoc Network, VANET, is a technology that uses moving vehicles as nodes in a network to create a mobile network. Each vehicle takes on the role of sender, receiver, and router to broadcast information to the network. This information is then used to ensure safe and free flow of traffic. Vehicles are equipped with some sort of radio interface called OnBoard Unit (OBU) that enables communication with other vehicles and with Road Side Units (RSU). Vehicles are also equipped with hardware that permits detailed position information such as Global Positioning System (GPS). Fixed RSUs, which are connected to the backbone network, must be in place to facilitate communication.

VANETs use Dedicated Short Range Communications (DSRC) which is a short to medium range communications service that was developed to support Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communications. Different standards (IEEE 802.11p, Wireless Access in Vehicular Environments (WAVE), and IEEE 1609) have been developed for VANETs.  These standards form the basis for deployment of VANETs and their communications [26,27]. Figure 1 shows a snapshot of a VANET where vehicles communicate with each other and with road side units.

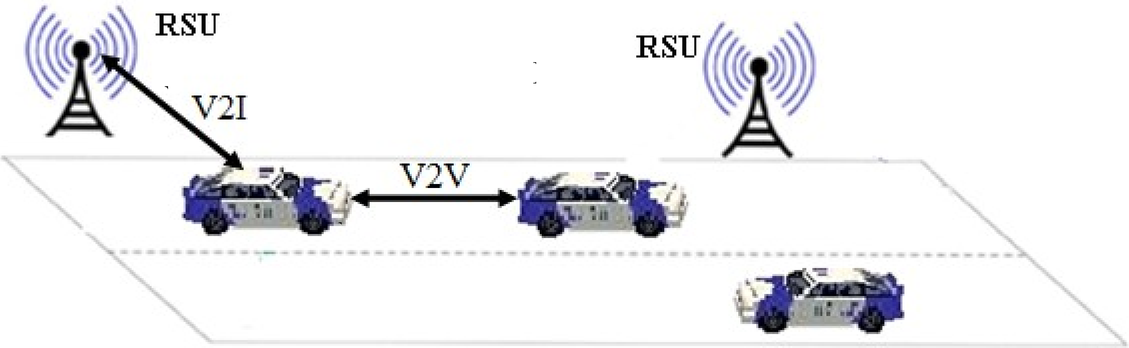


Figure 2: VANETs architecture

One of the mechanisms used for Authentication in VANETS is digital signature. Because of the large number of network members and variable connectivity to authentication servers, a Public Key Infrastructure (PKI) is used for authentication where each vehicle would be provided with a public/private key pair. Public keys are used as pseudonyms and are changed frequently for security and privacy issues [28,29].

### 2.2.2 Properties of VANETs Architectures

Due to the sensitivity of location data and its power to reveal real identities of users using data mining tools, different privacy preserving and secure techniques have been proposed. An important technique is the use of temporary public/private key pairs for authentication. These pairs form temporary identities for vehicles; hence they are used as pseudonyms. These pseudonyms are changed from time to time. This, in turn, is expected to hide the complete trajectory of a particular vehicle [21,30]. However, it is sometimes possible to link two pseudonyms and hence to link the corresponding locations updates. One way to tackle with this problem is to use mix zones. A mix zone is a region where – upon entrance - vehicles change their pseudonyms together, the new pseudonyms are mixed together and linking old and new pseudonyms becomes more difficult [29,31].

Generally, VANETs concentrate on anonymous access for users and do not deal efficiently with privacy of location data itself, i.e. privacy issues related to location points. Location data without identities contain sensitive information that may lead to the disclosure of the user's real identity. Because of the vehicle to vehicle (V2V) communication in VANETs, complex security protocols should be implemented to detect and prevent collusion among vehicles and other types of attacks [32]. Unfortunately, preventing V2V communication requires deployment of a very large number of RSUs to cover the entire region. This is very expensive and needs much more time to be done.

### 2.2.3 Recent Deployments of VANETs

Different trials of deploying VANETs were done in U.S.A, Europe, and Japan [19]. There are many national and international projects supported by government, industry, and academia devoted to these field trials. These include consortia such as the Vehicle Safety Consortium (VSC) in the U.S.A., Car-to-Car Communications Consortium (C2CCC) sponsored by the European Union, and the Advanced Safety Vehicle Program (ASV) in Japan. However, these deployments are relatively restricted in terms of services and geographical coverage. A full VANET deployment requires installation of new infrastructure, which is hard to be globally achieved in the next 10 years. Recently, the use of collaborative traffic monitoring systems that utilizes existing communication infrastructures is more promising such as [24,25,33].

### 2.3 Utilizing Existing Communication Infrastructures

EI approach is based on the utilization of existing underlying infrastructure such as cellular and wireless networks to set up the CTM system. The architecture is a client server architecture where client sends her information to the traffic server and gets a complete overview of traffic from that server. The client is assumed to have a positioning device such as GPS receiver to calculate her location, and a mobile communication device to communicate with the traffic server [3,23,24,33,34]. Most recent mobile phones are equipped with GPS and with many wireless communication capabilities such as GSM, Wi-Fi, GPRS, EDGE, etc. Figure 2 shows the general architecture of EI CTM systems.

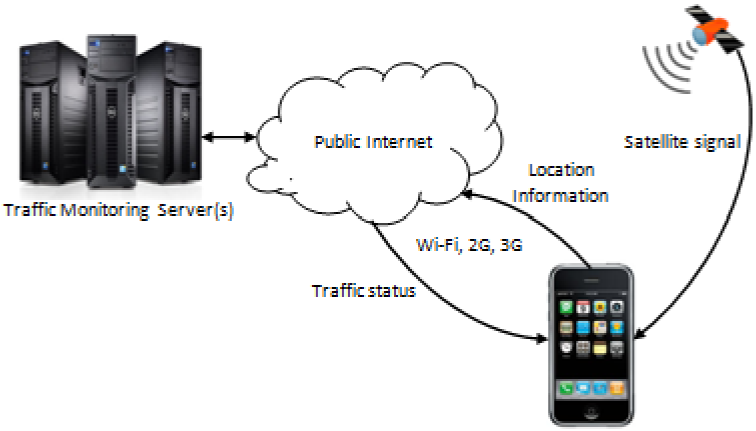


Figure 3: EI communication infrastructure

Utilizing existing communication infrastructures accelerates the development of traffic monitoring system because there will be no need for new communication infrastructure deployments.

Anonymous access with EI systems is limited to the use of a nickname; all location updates of a particular user are associated with her nickname. Some systems allow users to appear as anonymous on their live maps; however, this does not protect the privacy of that user. Also modern EI approaches for CTM systems do not support identity revealing for law enforcement purposes. This reduces the adoption of these systems by related authorities.

### 2.3.1 Properties of EI Architectures

Existing EI architectures depend on a Trusted Third Party, TTP, to protect user's privacy. TTP may breach user's privacy by revealing her location information. Different systems have been proposed to mitigate this full trust. A popular proposed solution is called *cloaking*. Cloaking means hiding the real data (location, time) with data of other objects by expanding exact location or time values to values where *k* other objects have [35,36,37]. Cloaking generates better privacy levels with less accurate data. However, cloaking still requires a trusted third party to calculate the boundaries of the cloaked region. The user sends her location to a cloaking server (sometimes called an anonymizer); the latter calculates the coordinates of the region where *k* users exist including the applying user. The user, then, replaces her exact location with these coordinates. There are different variations of cloaking; however, they still depend on a TTP which is not preferable for privacy issues.

### 2.3.2 EI Recent Deployments

One popular EI CTM system is called WAZE (waze.com). WAZE was founded first in Israel in 2006; now, it is being used in the USA and in some European countries. WAZE is a free system and requires a new user to register with her email address and then she receives her password via an SMS to her mobile phone. The system has evolved rapidly and it collects data from registered drivers. The system requires users to be connected to the Internet via some communication technology such as 3G, Wi-Fi, etc.

In WAZE, users authenticate using a user name and a password, users are allowed to use nick names for their activities. Although WAZE is becoming more popular, there are privacy risks associated with using this system:

1. WAZE users authenticate using a permanent user name and password. Misbehaving users may abuse the system by providing their user name and password to others. Imagine a CTM user recording a path in Istanbul, and after five minutes, the same user name recorded a path in Ankara. This abuse may affect the accuracy of the system.
2. Users are allowed to make changes on the map by recording a new track or changing a name of a place. WAZE cannot validate suggested updates/changes. It also does not protect privacy of others. Consider a WAZER who recorded the name of his neighbor on the location of his neighbor’s house, and his neighbor does not want to disclose his address to public. This violates the neighbor’s privacy and should not be allowed. Or at least liability issues have to be executed and the one who violates should be responsible for that violation.
3. By the use of nick name of a person, it is very easy to track all places that this user visited. Suppose a user with nick name X is using WAZE, all location information she sends to the server are saved with her nick name. So, by simply searching the database for that nick name, all her location information will be available without having to mine them. This violates her privacy and may reveal her real identity too.

Therefore it is necessary enhance WAZE with privacy enhancing mechanisms to protect users privacy.

Google provides live traffic reports included within the Google maps interface. This service is not available for all cities. It is available in the USA, Canada, and some European cities as well. The traffic reports are updated every five to ten minutes and are currently available on Google Maps and Google Maps for mobile, and Google Maps Navigation. The live reports help to avoid heavily congested roads and they also offer alternative routes.

The traffic reports are useful for people who want to plan their routes ahead. Using previously stored traffic information, you select the time, date, location and the traffic reports will provide a trend in traffic levels. Thereby enabling users to plan ahead and avoid heavily congested roads.

Google traffic data come directly from local highway authorities, and from GPS enabled phones that use Google Maps with the location tracking feature enabled [38]. As users move around a city, Google can see how well traffic is flowing along any road and will update its live traffic data accordingly. Due to the lack of available traffic data, Google live traffic reports mainly covers main roads and highways.

YANDEX[3] is a Russian Internet company that operates the Yandex search engine in Russia. It also develops a number of Internet-based services and products including Yandex Maps and Yandex Traffic. Yandex Traffic shows the picture of the current traffic conditions in a city. It gathers information from different sources, analyses this data, and maps the results on the city's map on Yandex Maps. Yandex users may benefit from traffic reports and avoid congestions. It is worth mentioning that Yandex works now in Turkey and provides traffic information for cities like Istanbul and Ankara.

## 2.4 Location Update Mechanism in CTM Systems

CTM system requires users to update their location information at the traffic server from time to time so that the traffic server will be aware of traffic conditions. Different update mechanisms have been proposed in the literature. These update mechanisms vary according to their update frequencies and time gaps between successive updates. Here, we briefly describe these mechanisms and their pros and cons.

### 2.4.1 Periodical Update Mechanism

In periodical update mechanism, location information is updated periodically at fixed time intervals [39]. By carefully fixing the interval between two successive updates, periodical update mechanism produces the best data in terms of quality. However, this

[3] http://www.yandex.ru/

mechanism suffers from the high probability of linking location updates of a particular moving object [30]. This high probability of linking stems from the periodic location update pattern that facilitates prediction of the time and location of the next update according to current time and speed. This in turn may lead to a partial or even total trajectory disclosure of a particular moving object.

By knowing current position and speed of a particular vehicle at time $t_1$, the expected location at time $t_2$ can be calculated by calculating the distance travelled by that vehicle during the time interval $t_2$-$t_1$. The distance is calculated as ($t_2$-$t_1$)$*$speed at time $t_1$ [29]. This model assumes a fixed speed interval over the time period $t_2$-$t_1$. There are better probabilistic models that incorporate the average speed of the route rather than vehicle's previous speed and use some probabilistic models to link updates according to their probabilities of occurrence [21,31].

### 2.4.2 Conditional Update Mechanism

Another location update mechanism suggests to update only if a vehicle crosses a boundary [25,34]. This mechanism is called conditional, i.e. location is updated if a condition is met. The condition is the cross of a predefined boundary. So if a moving object crosses a boundary, then the vehicle should update her location. These boundaries are pre-selected and distributed to users. The selection of these boundaries should be done carefully to ensure well coverage and better privacy.

This mechanism enables monitoring traffic only around these boundaries and ignores other regions. Besides, if a prior knowledge of these boundaries is obtained, then linking of location updates will become an easy task. Once two boundaries are compromised, an adversary may find out the distance between these boundaries, she also can find an estimation of the speed between these boundaries. This will help her to calculate the time of the next location update. So the problem becomes similar to the periodic update mechanism. Another important drawback of this mechanism is its dependency on a trusted third party to generate and distribute these boundaries.

### 2.4.3 Silent Period Mechanism

Silent period update mechanism suggests that vehicles use random periods of time between their successive updates. If a vehicle sends a location update at time $t_1$, then the next update will be $t_1+t_{rand}$, where $t_{rand}$ is a random number sampled randomly from a distribution. This random period is called silent period [21,40].

Because of the lack of periodicity in location updates, silent period makes it difficult to link updates of a particular user. However, probabilistic models may still be able to do that with high confidence. Using silent period update mechanism will not make it possible to catch all traffic conditions in the entire region meaning that it will degrade the feasibility of CTM system.

## 2.5 How do CTM Systems Become Privacy Invasive

CTM systems depend on the collection of users' location data to build and develop the system databases required for traffic monitoring and map generation. This data are stored in the database as they are collected. The data are then used for creating traffic maps and reports.

CTM systems are self-positioning by which a user sends her location to CTM server. Such systems protect privacy if CTM server is a trusted party that does not intentionally or unintentionally share the data with other parties. Unfortunately, this may not be the case. Different privacy attacks may be applied to such systems such as:

1. Profiling the user's behavior: utilizing GPS tracking data of a particular user, and with some data mining tools, a user may be profiled to a given group according to her preferences and activities [2].
2. User tracking and identification: with some data mining tools, spatio-temporal data can be used to cluster users and then infer their real identities according to their routes [11,34].

Location data are sensitive data and may severely harm the user's privacy. The above attacks are general attacks that many other scenarios can be listed below them. The severity of the harm depends on the sensitivity of the disclosed information and the related person.

Political activities, health, ethics, security records, and other information can be extracted from location data. This information may be of high sensitivity of a particular person, and the disclosure of such information may lead to harmful consequences.

## 2.6 Related Work

There are two stages of our proposed CTM system; the first stage is building the anonymous access system where the second is designing the location update mechanism. In this Section, we address the related work for both stages. For the sake of simplicity for readers, we separated the related work.

### 2.6.1 Related Work for Anonymous Access and Pseudonyms Systems

There are two approaches in the literature that address anonymous service access. The first approach is called *anonymous blacklisting* (a.k.a. *anonymous revocation*). This approach allows revocation of misbehaved users without revealing their real identities. It also maintains previous anonymity for even abusive users. The second approach is called *revocable anonymity*. In this approach, abusive users are revoked and their real identities are revealed as well.

In anonymous blacklisting, various Trusted Third Party (TTP) schemes have been proposed. These schemes assume a level of trust among parties. The first anonymous TTP blacklisting scheme to appear in the literature was proposed by Johnson et al. and called Nymble [41]. Nymble constructs unlinkable authentication token sequences using hash chains. A pair of TTPs, the Nymble manager and the pseudonym manager, help the service providers to link future tokens from abusive users so their access can be blocked. Unfortunately, these TTPs can easily collude to de-anonymize any user.

Nymbler [42], BNymble [43], and Jack [44] are similar schemes that have been proposed with some performance enhancements on the base scheme Nymble. With the aim to force an agreement between users and service providers, Schwartz et al. have proposed a contractual anonymity system [45]. In this system, a user is de-anonymized if she breaches the contract with the service provider. This system still depends on a TTP.

BLAC [46], EPID [47], and PEREA [48] are anonymous service access systems in which abusive users are revoked without contacting a TTP. In these schemes, service providers simply add authentication tokens associated with misuse to a blacklist. When a user produces a new authentication token, she must then prove that each token on the blacklist is not linked to her new token. This becomes harder to do as the number of users and revoked tokens increase.

Revocable anonymity systems (the second approach) generally depend on cryptography to generate and verify anonymous identities that are sometimes called *pseudonyms*. The concept of pseudonyms was introduced by Chaum [49] as a way of allowing users to communicate with different organizations using temporary identities. Later, Chaum and Evertse [50] have developed a model for pseudonym systems. They have presented their system as an RSA-based implementation. Their scheme relies on a TTP to sign all credentials.

The use of TTP to sign credentials and reveal real identities of pseudonyms was employed by many service providing systems such as VANETs (Vehicular Ad hoc Networks) described in [30,39,50,51]. In these systems, the authors propose the use of pseudonyms to access the service anonymously while maintaining the ability to revoke abusive pseudonyms by revealing their real identities. It has been shown that pseudonyms may be linked and anonymity may be revealed as well [29]. To overcome the latter problem, the concept of mix zones has been proposed by Freduiger et. al. [51]. A mix zone is an area where many vehicles change their pseudonyms at the same time, causing new pseudonyms to mix together and making it difficult to link old and new pseudonyms. A similar approach has been proposed by Lu et. al. [52] utilizing the so-called social spots to create a mix zone. A study of the effect of non-cooperating users at mix zones was done by Freudiger et. al. [53]; they also proposed a protocol that deals with non-cooperative users.

Group signature schemes, such as [54,55,56,57], have been widely used for both anonymous blacklisting and revocable anonymity systems. Based on group signature features, an open authority can revoke abusive users and may reveal their real identities. Figure 3 gives a categorization of anonymous access systems.

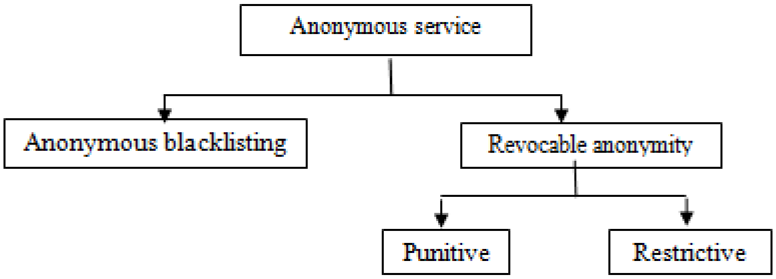


Figure 4: Categories of anonymous access systems

All previous systems are either punitive in a way that they allow TTPs to reveal past and future anonymity of a particular user, or they are restrictive in a way that they allow revealing only current pseudonym. Each previously described pseudonym system fits to a particular service providing systems and may not fit to others; this depends on the nature of the service provider. In many applications, such as VANETS, de-anonymization is sometimes required for a period of time that may span more than the lifecycle of a single pseudonym. Hence, there is a necessity for a flexible system that maintains anonymity, distributes trust, and enables fair identity de-anonymization. In this paper, we propose a collusion resistant pseudonym providing system that addresses these issues.

### 2.6.2 Related Work for CTM

A secure dedicated infrastructure (DI) system architecture has been proposed by Raya et al. [58]. In this scheme, the authors propose the use of *pseudonyms* for identity hiding. Pseudonyms are temporary identifiers that expire after their use. Although pseudonyms were temporary identities, it has been shown that it is possible to track pseudonyms changes and disclose real identities by using some probabilistic models [29].

To overcome the problem of linking pseudonyms, the concept of mix zones has been proposed [51]. Mix zones are hidden areas where users can changes their pseudonyms together without being linked. In this way, the pseudonym change is not monitored, and hence pseudonyms will not be linked. Pseudonyms and mix zones have been proposed to be used in Dedicated Infrastructures (VANETs). However, DI approach is still under research [2,28,39] and no important real deployments exist.

Different online privacy preserving approaches have been proposed for location data. One major approach is *cloaking*, either time, space, or both [25,35,37]. Cloaking is a process of generalization, where time or space is expanded so that a *k*-anonymity level is met, *k*-anonymity refers to the state of being anonymous among other *k* objects [59]. Cloaking gains a guaranteed *k*-anonymity level on the account of the quality of the data. As a drawback, all cloaking techniques rely on either a trusted third party that determines the boundaries of the cloaked region [36], or on collaboration among users to update their locations together [37]. The latter relies on direct communication between group members, and requires trust between them.

Hoh and Gruteser provided the concept of Virtual Trip Lines (VTL) [25]. A VTL is a geographic boundary that is supplied to the client software, and a vehicle must update her location upon the cross of that boundary. The system fails to capture traffic conditions apart from VTL regions, because vehicles will not update their locations outside VTL regions. Besides, if some VTL lines are compromised then it will be easy to link pseudonyms at these compromised VTLs. The very much effort done on the choice of VTLs and their distribution to users makes the system impractical for a larger number of users.

In [36], the authors use an anonymization server that anonymizes user locations using location cloaking, where location information is perturbed by either spatial range called spatial cloaking, or temporal range called temporal cloaking. Thus, exact location information is hidden among a range of temporal and spatial coordinates. It guarantees *k*-anonymity in both time and space dimensions. But it still relies on a trusted third party and also degrades the quality of the data. In [33], the authors suggest a 2 way cloaking mechanism, the user sends her cloaked location to an anonymization server, the latter returns a cloaking rectangle that have *k* other users. Cloaking relies on a trusted third party for calculating the safe region; it also reduces the data quality by generalizing location into the safe region. A similar approach was used by [37] where users communicate together to form a safe region without communicating with a trusted third party, in the latter approach users are assumed to be honest and trust each other to calculate the safe region.

In [1], we studied the challenges of a privacy preserving collaborative traffic monitoring systems utilizing existing infrastructure.

## 2.7 Summary

In this chapter, we provided a background and an intensive survey of related work. We also listed the different approaches used in collaborative traffic monitoring systems. As well as describing different location update mechanisms used in these systems. In the next chapter, we provide an introduction about our proposed Privacy Aware Collaborative Traffic Monitoring System, PA-CTM.

# 3. OUR PROPOSED PRIVACY AWARE COLLABORATIVE TRAFFIC MONITORING SYSTEM DESIGN

In this chapter, we describe the general design and architecture issues of our PA-CTM system. Our design is divided into two stages. The first stage is the design of the pseudonym signing system that signs pseudonyms required for anonymous access to the traffic server. The second stage is the design of location update and pseudonym change mechanism that determines when to update a user's location and when to change the used pseudonym. Here, we describe the general design and flow of our PA-CTM, details about the first stage and the second stage are done in Chapter 4 and Chapter 6 respectively.

## 3.1 PA-CTM Architecture

The architecture of PA-CTM is client server architecture. Users use their signed pseudonym to authenticate to traffic server for either updating their location information or querying current traffic status. The traffic server collects location information from different moving objects (identified by pseudonyms) and provides users with current traffic status. Figure 4 shows the general architecture of our PA-CTM system.

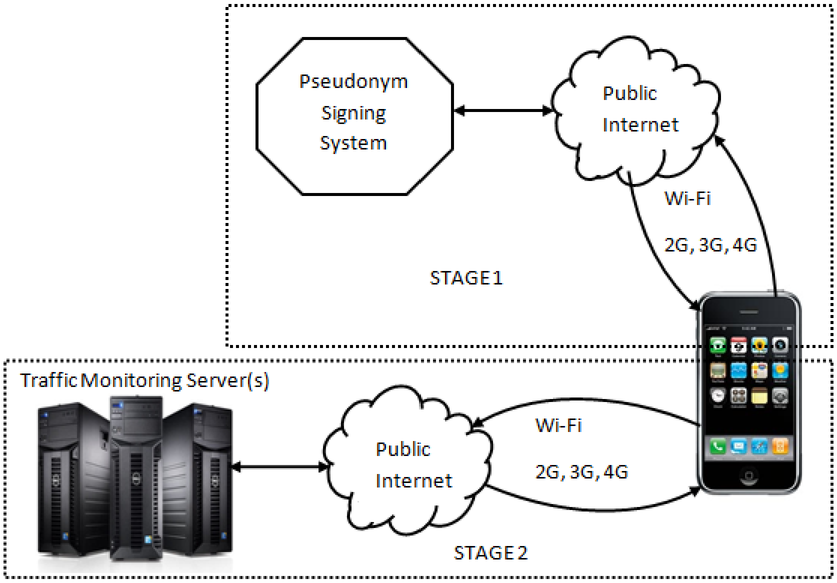


Figure 5: PA-CTM general architecture

The flow of PA-CTM is performed as follows:

1. The user generates a number of temporary identities called pseudonyms, and then she sends these pseudonyms to the pseudonym signer who in turn signs them and returns the signatures to the user. Details about authentication and signing pseudonyms are provided in Chapter 4.
2. The user can authenticate to the traffic server using one of her signed pseudonyms. Note that users are capable to change pseudonyms from time to time and hence to divide their real trajectory into smaller trajectories. This in turn makes it hard to link different pseudonyms for the aim of constructing the complete trajectory.
3. The traffic server verifies the signature of the Pseudonyms Signer and responds to the user accordingly. Verification of signature is done using the public key of pseudonyms signer.

When a user requires signing new pseudonyms, she simply generates new pseudonyms and then applies again to the pseudonym signer. For security issues, pseudonyms are valid for a predefined lifetime and then expire.

## 3.2 Pseudonym Signer

The main idea of using a pseudonym signer is to gain anonymous access to traffic server similar to that used in VANETs designs. To accomplish this task, a pseudonym signer should have the following properties:

- Requires registration
- Enables Revocation for misbehaving users
- Have a back door for identity revealing under law enforcement purposes
- Preserves privacy and distribute trust among different entities.

Registration is required for determining legitimate users and for other purposes such as billing of services provided to users. It is also required for restricting the use of the service to legitimate users only. Revocation is required for preventing misbehaving users from continuing using the services. CTM systems are related to cases where liability may be required; these cases include road accidents, robbery, etc. In these cases there might be some need for revealing real identity for law enforcement purposes. On the other hand, an adversary should not be able to link a particular pseudonym to a particular user, she also should not be able to link pseudonyms used by a particular user. Details about the design and properties of pseudonym signer are provided in Chapter 4.

## 3.3 Design of Location Update and Changing Pseudonyms

This stage aims to leverage privacy of users by enhancing the privacy level of their location data. Location information is sensitive and may reveal important information about users. The decision of when to change a pseudonym is very important and may significantly enhance privacy. Changing a pseudonym with other *k* users at the same time and region reduces the probability of linking old and new pseudonyms together, this in turn enhances privacy of users who have changed pseudonyms together. In VANETs, the region where vehicles change pseudonyms together is called a mix zone. Mix zones in VANETs depend on a trusted third party to establish these zones and distribute them to users.

In existing infrastructures, the privacy is enhanced by cloaking techniques, these techniques guarantee *k*-anonymity level, however they depend on trusted third parties for determining the cloaking region or time.

In our PA-CTM, we propose a design of a pseudonym change mechanism that behaves similar to mix zones in VANETs and leverage privacy of users. It also triggers a location update according to local parameters (speed and direction) and does not rely on a trusted third party.

## 3.4 Summary

In this chapter, we provided an introduction of our proposed PA-CTM. In the next chapter, we give detailed explanation of our collusion resistant pseudonym providing system, CoRPPS that will be used in building our privacy aware collaborative traffic monitoring system.

# 4. COLLUSION RESISTANT PSEUDONYM PROVIDING SYSTEM

In this Chapter, we provide the details of our Collusion Resistant Pseudonym Providing System, CoRPPS. These details include detailed design and flow, properties of CoRPPS including identity revealing for liability and revocation of misbehaving users, and resistance against attacks including collusion among entities and among users as well.

## 4.1 Introduction

As discussed in [13,14], the lack of privacy is the main hindrance for the success of a service providing system that requires user authentication. This encouraged service providers to develop a privacy preserving system that protects users' privacy. Most of these systems depend on the usage of temporary identities instead of real identities. These temporary identities are called *pseudonyms* [50].

Anonymous access systems, such as Tor [63] , and Crowds [64], allow users to connect anonymously to service providers by rerouting traffic through a number of network servers. Some service providers require the possibility of denying the service for abusive users. Using such anonymous access systems would cause denying the service to legitimate users as well. Thus there is a necessity for anonymous access systems with the possibility of revoking abusive users only.

Access control and revocation rules differ from one service provider to another according to the nature of the service provided. Some providers need only to revoke abusive users without revealing their real identities. Some requires revealing the real identity of the current pseudonyms without looking for the past pseudonyms (i.e. previous anonymity is guaranteed). In other applications such as traffic monitoring systems, current and part of previous anonymity revealing may be necessary due to law enforcement reasons.

Different anonymous access systems with different properties have been proposed in the literature [65,66]. These systems are either punitive [65] in a way they completely reveal previous anonymity, or restrictive [43] in a way that they only revoke future access without revealing real identities. Most of these systems, as [39,67], have been designed for a specific service provider and may not suit other services. Another drawback is that all systems that enforce identity revealing rely on a Trusted Third Party (TTP) to reveal the real identity. Misbehavior and/or collusion of TTP may lead to a severe privacy leakage, which has not been addressed adequately in the existing schemes.

In this chapter, we propose the design details of our Collusion Resistant Pseudonym Providing System, CoRPPS [15, 16], a novel pseudonym providing system. CoRPPS distributes trust among all system parties and resists against collusion among them to reveal the real identities of the users. In this way, CoRPPS ensures a level of anonymity for users served by a particular service provider. It also enables linking a particular pseudonym to its real identity for liability reasons only. Identity revealing is fair and does not reveal services other than the required case. By the term *liability* we mean disclosing real identity for law enforcement purposes in cases such as road accidents, robbery, etc. To the best of our knowledge, CoRPPS is the first work in pseudonym systems that address flexibility, identity revealing fairness, and collusion among all type of system parties. CoRPPS is *flexible* such that it fit into different services by adjusting its parameters..

## 4.2 CoRPPS Design

As a pseudonym providing system, CoRPPS should have the following properties that are required for such systems:

- Registration
- Revocation of misbehaving users
- Identity revealing for law enforcement purposes
- Privacy preserving and distributed trust

These properties are crucial for a pseudonyms providing system. The success of such systems relies heavily on these properties. In addition to these properties, we added a new feature to our design; this feature is *flexibility*. Flexibility means that CoRPPS can fit different application including traffic monitoring system; details about flexibility are provided in Section 4.6.1. According to the above mentioned properties, we design CoRPPS to be composed of the five functional units listed below:

1. Registration Authority, $RA$: $RA$ is responsible for the process of users' registration.
2. Users: In our Privacy Aware Collaborative Traffic Monitoring, PA-CTM, users are vehicles that are subscribed to the system and have the rights to use it. It does worth mentioning here that CoRPPS is a multipurpose system that can be tuned for different service providers including traffic monitoring.
3. Authentication Servers, $AS$s: The aim of using multiple $AS$s is to split user authentication among a group of authentication servers. This in turn prevents a particular authentication server from being able to link a temporary identity to a particular user.
4. Pseudonym Signer, $PS$: $PS$ signs user's pseudonyms using her private key. This signature can be verified later by the Service Provider using the public key of $PS$.
5. Service Provider, $SP$: In PA-CTM, $SP$ is the traffic monitoring system. $SP$ can check the legitimacy of a pseudonym by verifying the signature of $PS$ using $PS$'s public key.

These units (shown in Figure 5) communicate together to form the general flow of CoRPPS. The process of putting CoRPPS into operation requires the execution of the following steps; these steps are detailed in Section 4.5 and are shown in Figure 5 as well.

1. Initial setup: The aim of this stage is to prepare CoRPPS units for registering users and providing services to them.
2. Registration: Users register to the registration authority using their identification information

3. Authentication and ticket acquisition: Users apply to a predetermined number of authentication servers, $AS$s, to get *tickets*[4]. These tickets are used by pseudonym signer to check the legitimacy of the pseudonyms signing request.
4. Signing pseudonyms: users send *tickets* and a set of pseudonyms to pseudonyms signer who in turn verifies the correctness of the tickets and sign pseudonyms.
5. Using the service: Users use the service by authenticating themselves using their signed *pseudonyms*. Service provider verifies the signature of pseudonym signer over these pseudonyms and proceeds with request accordingly.

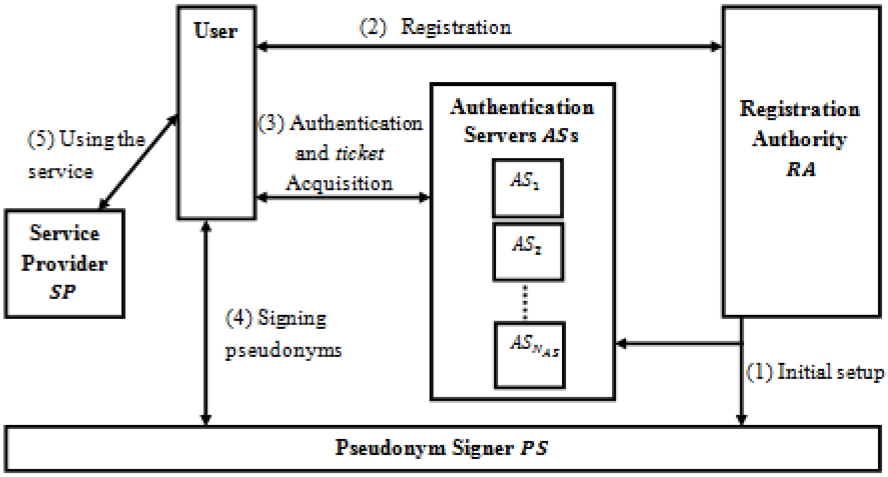


Figure 6: CoRPPS design

## 4.3 Assumptions and Threat Model

In CoRPPS design, we assume that all communications among CoRPPS entities are secured using SSL (Secure Socket Layer) or another transport layer security protocol.

The Pseudonym Signer, $PS$, has a public-private key pair and uses this to sign *pseudonyms*. The Service Provider, $SP$, knows the public key of $PS$.

[4] A *ticket* can generally be described as hidden information to be sent to pseudonym signer through the user. A more detailed description is provided in Section 4.4.4.

Users are assumed to be semi-honest such that they follow the protocols properly and do not block the continuity of CoRPPS; however, they are curious and try to link *pseudonyms* to the real identities of particular users.

On top of this curiosity, parties may collude by exchanging secret or critical information that they possess in order to reveal real identities of *pseudonyms*, and hence try to breach privacy of a *pseudonym*'s holder. Collusion may occur between any two or more parties of CoRPPS except users who are assumed to collude only together.

The security of CoRPPS does not depend on an ultimately trusted entity. Instead the trust is split among multiple entities and our design resists against collusion among them.

It is a general requirement for most pseudonym system to have a backdoor for identity revealing to be used by law enforcement units when needed for legal and liability cases. CoRPPS also supports this feature. Actually such a feature contradicts with user privacy, so a careful design is needed. In our CoRPPS design, in order to maximize the privacy of the users, all trusted system entities must collaborate together in order to reveal the real identity of a user who used a particular service. In other words, collusion among all trusted entities is not considered as a threat, and this fact is by design.

## 4.4 CoRPPS Basic Building Blocks

The basic building blocks of CoRPPS are *tokens* and *token pool*, *counter*, *verification code*, *tickets*, and *pseudonyms*. The following subsections provide detailed information about each of them.

### 4.4.1 Tokens and Token Pool

Tokens are temporary anonymous identifiers which help $PS$ to verify that a user is a genuine user. Tokens are generated by the registration authority, $RA$, to be used by the authentication servers, $AS$s, to generate users' *tickets*. $RA$ generates a token pool at the setup phase and sends this pool to all $AS$s, and to the pseudonym signer, $PS$. $AS$s use the token pool to randomly select a token with replacement and use it to generate the users' *tickets* as described in Section 4.5.3.

The random token selection process is *with replacement*, meaning that a particular token can be used several times for the same or different users; this is one of the main enablers of anonymity in our system. On the other hand, $PS$ uses the token pool to validate the process of signing users' *pseudonyms*. New token pool is to be generated when CoRPPS collision probability exceeds a threshold; for more details about collision probability, please refer to Chapter 5.

### 4.4.2 Counter

In CoRPPS, a particular user is assigned a group of Authentications Servers, $AS$s, during registration and she always talks to this group of $AS$s to obtain *tickets*. A particular user $U$ and her corresponding group of $AS$s maintain a synchronized counter, $Ctr_U$. $Ctr_U$ holds the number of times the user $U$ has applied to $AS$s for *tickets* (i.e. it is a session counter). Hence it is incremented in both $AS$s side and user side when the user $U$ gets her *tickets*.

$Ctr_U$ is limited by the value $Ctr_{max}$ in order to enforce the users to use the system efficiently and fairly. Once users consume their counter values, CoRPPS must be restarted with new token pool and the counter values of all users are set to 0. $Ctr_U$ is used to calculate the verification code and its main function is to have a different verification codes for the same user at each session. This is very important to provide unlinkability among different sessions.

### 4.4.3 Verification Code

Verification code, $v$, is a value calculated at each authentication server whenever a user applies for *tickets*. This code is unique for a particular user, $ID_U$, a particular group, $GID_U$, and a particular $Ctr_U$. Each time user $U$ applies to an $AS$ for a *ticket*, $AS$ calculates $v$ as $v = \text{hash}\left(ID_U||Ctr_U||GID_U\right) \; mod \; V_{max}$ ,where $ID_U$ is the identity of the user, $Ctr_U$ is the current counter value of user $U$, $GID_U$ is the group identity of the $AS$s corresponding to that user, and $V_{max}$ is the upper bound of $v$ values.

In setup phase, authentication servers, $AS$s, are organized by registration authority into groups of $g$ $AS$s per group. Each group is given an identity called group identity, $GID$. Then, in registration phase, users are distributed among these groups and each user is only allowed to apply to her assigned group of $AS$s, $GID_U$. All $AS$s of the same group generate the same verification code, $v$, for the same user identity, $ID_U$, and the same counter value of that user, $Ctr_U$. $AS$ uses $v$ to generate the *ticket* that will be sent to the applying user. Pseudonym signer, $PS$, uses $v$ to verify that the received $g$ *tickets* from a particular user actually belong to this user, if these *tickets* bear the same $v$ values.

### 4.4.4 Tickets

*Tickets* are pieces of encrypted information generated by authentication servers, $AS$s, and sent to the Pseudonyms Signer, $PS$, through the user. *Tickets* are encrypted using a symmetric encryption key $K$ shared between $PS$ and $AS$s. A *ticket* is generated by an $AS$ as $ticket = E_K(t||v)$, where $t$ is a randomly selected token from token pool, and $v$ is the corresponding verification code. *Tickets* are used by $PS$ to check that tokens are valid and that *tickets* have the same verification code, which means that they are generated for the same user with the same *counter* value. The aim of encryption is to hide the *ticket* content from users so that they cannot misuse this information to cheat on the system via collusion among themselves.

*Tickets* are used by $PS$ to check that tokens are valid and that *tickets* have the same verification code, which means that they are generated for the same user with the same counter value.

### 4.4.5 Pseudonyms

*Pseudonyms* are temporary identities used by users to apply to the service provider for a service. Users generate *pseudonyms* as random values and send them to the

Pseudonym Signer, $PS$, at which they are signed. After that, the user submits a signed *pseudonym* to apply for a service. In the notations, we use $P_U^i$ to denote the *i*th *pseudonym* of user $U$. Although, the notation implies that the user identity and ordering information is in the *pseudonym* structure, it is actually not the fact. User and order information is used in the notation for the sake of clarity of the explanation. Actually, the *pseudonyms* are random values and do not carry any user information in themselves. Moreover they do not follow a particular sequence. Otherwise, a particular user's *pseudonyms* may be linked according to that sequence which may cause a dangerous privacy breach.

## 4.5 CoRPPS Flow

In this subsection, we explain, in detail, all five stages shown in Figure 5 (initial setup, registration, authentication and *ticket* acquisition, signing pseudonyms, and using the service). These stages form the entire flow of CoRPPS and achieve the aim of CoRPPS design, i.e. providing *pseudonyms* while maintaining a high level of resistance against collusion attacks.

### 4.5.1 Initial Setup

The main role of this stage is carried out by the registration authority, $RA$. This step is explained in Figure 6. During the initial setup, $RA$ generates a token pool and distributes it to all authentication servers, $AS$s, and to the pseudonym signer, $PS$. The size of the token pool is denoted by $N_t$. $RA$ groups $AS$s in equally sized groups. The size of a each group is denoted by $g$. Selection of $g$ value is a tradeoff which is going to be discussed in Chapter 5. The total number of groups is simply calculated as $\binom{N_{AS}}{g}$.

Each group is given a group identity $GID$. This identity is sent to the corresponding $AS$s to be used in later stages of CoRPPS. The main benefit of grouping is to provide the resistance against collusion among $AS$s, as will be discussed later. Note that a particular $AS$ may belong to more than one group, and hence may own different $GID$ values for different groups it belongs to.

The final step in CoRPPS setup is generating a symmetric encryption key $K$ by $RA$ and sending it to all $AS$s and to pseudonym signer. $K$ is used to encrypt and decrypt tickets between $AS$s and $PS$. In the setup phase all the exchanges are assumed to be performed over secure offline channels

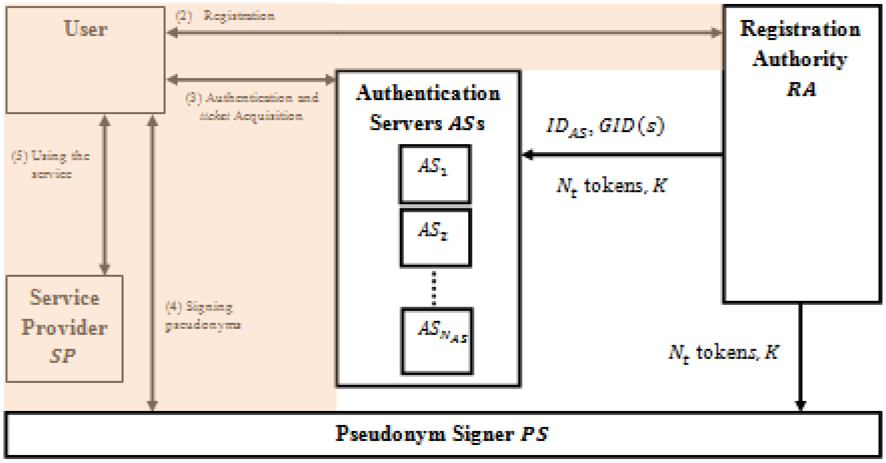


Figure 7: Initial setup

### 4.5.2 Registration

Each user in CoRPPS is to be registered before getting the *pseudonyms* signed and using the services. The main task in registration is to specify the group of $AS$s with which a particular user interacts and make the necessary setup for these interactions. These steps for user registration are shown in Figure 7.

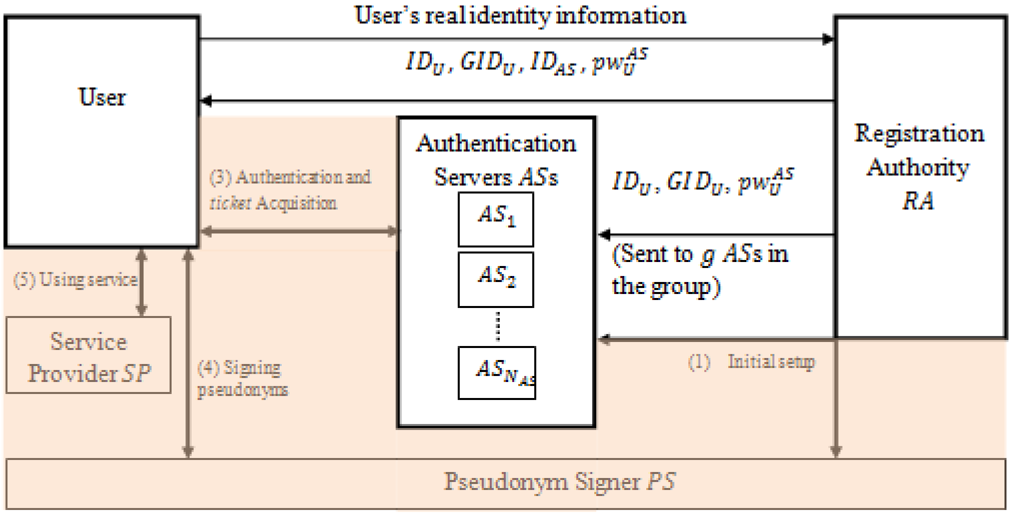

User's real identity information
$ID_U, GID_U, ID_{AS}, pw_U^{AS}$
User
Registration Authority $RA$
Authentication Servers $AS$s
$AS_1$
$AS_2$
$AS_{N_{AS}}$
(3) Authentication and ticket Acquisition
$ID_U, GID_U, pw_U^{AS}$
(Sent to $g$ $AS$s in the group)
(5) Using service
Service Provider $SP$
(4) Signing pseudonyms
(1) Initial setup
Pseudonym Signer $PS$

Figure 8: Registration

For a user to register in CoRPPS, she first sends her real identification information to $RA$. $RA$, then, generates a random identity, $ID_U$ and $g$ passwords, one for each $AS$ in the group, for that user, and sends them back to her. The identity $ID_U$ has been selected randomly and does not include any indication about the real identity of the user; thus, disclosure of $ID_U$ does not cause revealing the real identity.

For user $U$, $RA$ generates $g$ distinct passwords, denoted as $pw_U^{AS}$, each of the is to be shared between user $U$ and an $AS$ in her group. These passwords are needed for the challenge-response authentication protocol between $AS$s and the user before the ticket generation process, as will be discussed in Section 4.5.3.

Moreover, $RA$ assigns the user $U$ to a particular group of $AS$s selected randomly from the available groups established in the setup phase. Later $RA$ sends $ID_U$, the user's group identity, $GID_U$, and $pw_U^{AS}$ to the corresponding $AS$s. $RA$ also sends the user the identities of $AS$s that belong to the user's assigned group. This would enable the user to communicate with her assigned $AS$s and would enable the corresponding $AS$s to authenticate that user using her $ID_U$ and $pw_U^{AS}$.

Note that $AS$s other than group members are not able to authenticate the user because they do not possess $ID_U$ and $pw_U^{AS}$, and each $AS$ in a particular group has a different shared $pw_U^{AS}$ for the same user $U$.

$RA$ maintains records of user identities, $ID_U$ , the identities of $AS$s assigned to that group, $ID_{AS}$, and the group identities of the users, $GID_U$. These records are used for revocation and identity revealing which are going to be discussed in Sections 4.6.2 and 4.6.3 respectively.

### 4.5.3 Authentication and Ticket Generation

Once the user registers, she can apply to her group of $AS$s to get authenticated and obtain *tickets*, which, in turn, are used to obtain signed *pseudonyms*. A user can run this process several times during her lifetime; however in this section, a single run of the process is detailed. Figure 8 shows the complete authentication and *ticket* generation stage.

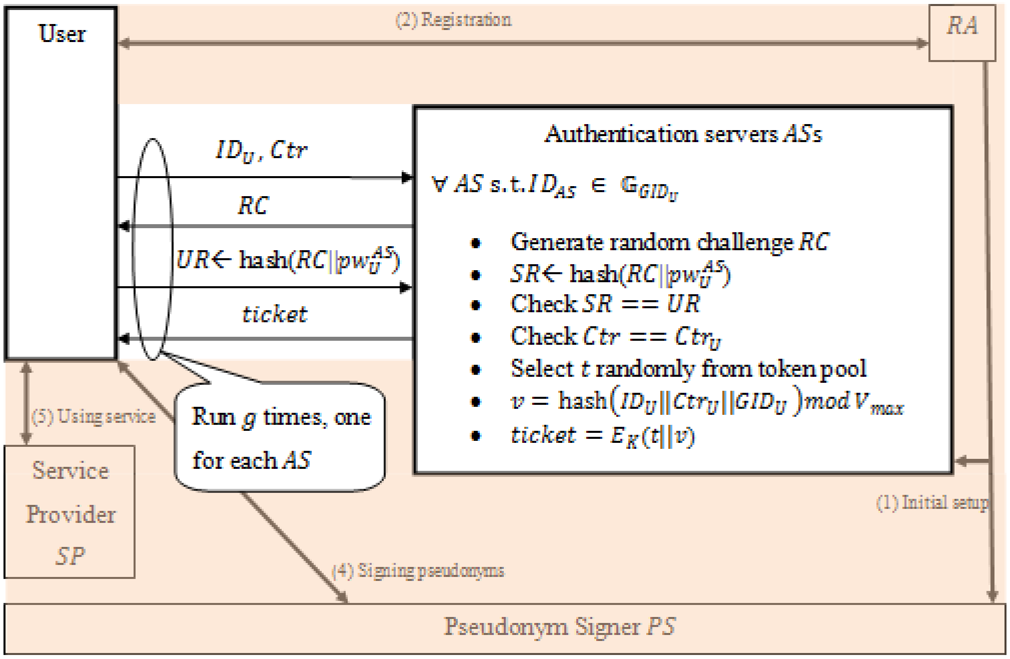

User
(2) Registration
$RA$
$ID_U, Ctr$
$RC$
$UR \leftarrow$ hash$(RC||pw_U^{AS})$
$ticket$
Authentication servers $AS$s
$\forall$ $AS$ s.t. $ID_{AS} \in \mathbb{G}_{GID_U}$
• Generate random challenge $RC$
• $SR \leftarrow$ hash$(RC||pw_U^{AS})$
• Check $SR == UR$
• Check $Ctr == Ctr_U$
• Select $t$ randomly from token pool
• $v =$ hash$(ID_U||Ctr_U||GID_U)$ $mod$ $V_{max}$
• $ticket = E_K(t||v)$
(5) Using service
Run $g$ times, one for each $AS$
Service Provider $SP$
(1) Initial setup
(4) Signing pseudonyms
Pseudonym Signer $PS$

Figure 9: Authentication and ticket generation

The user $U$ sends her $ID_U$, and $Ctr_U$ to all authentication servers, $AS$s, belonging to her assigned group, $\mathbb{G}_{GID_U}$. Each of the $AS$s in the group authenticates the user using a simple challenge-response protocol by first checking $ID_U$, and then sending a random challenge, $RC$, to the user. The user calculates her response $UR =$ hash$(RC \mid\mid pw_U^{AS})$ and sends it back to the authenticating $AS$. This $AS$ also calculates the response and compares the received response with the calculated one; if they are equal, then the user is authenticated.

After the authentication, $AS$ validates $Ctr$ value by simply comparing the $Ctr$ value sent by user with its $Ctr_U$ value; they should be equal. After the successful authentication and $Ctr$ validation, each $AS$ first calculates the verification code, $v = \text{hash}\big(ID_U||Ctr_U||GID_U\ \big)\text{mod}\, V_{max}$. Then, $AS$ selects a token, $t$, from token pool randomly and with replacement in order to generate a $ticket = E_K(t||v)$. The *ticket* is sent to the user. Since there are $g$ $AS$s in a group, the user receives $g$ *tickets* that will be used to get her *pseudonyms* signed by the pseudonym signer, $PS$. After the ticket generation, counter values of the user and the corresponding $AS$s are incremented by one.

Each $AS$ keeps the records of $(ID_U, ticket)$ pairs for all issued tickets. These records are used for identity revealing and revocation purposes as will be detailed in Section 3.4. Since the *tickets* are encrypted with $K$ and the users do not know it, *ticket* contents cannot be seen by the users. Moreover, *tickets* cannot be modified by the user either since if modified, decryption by the $PS$ in the next stage would not yield the correct $t||v$ form.

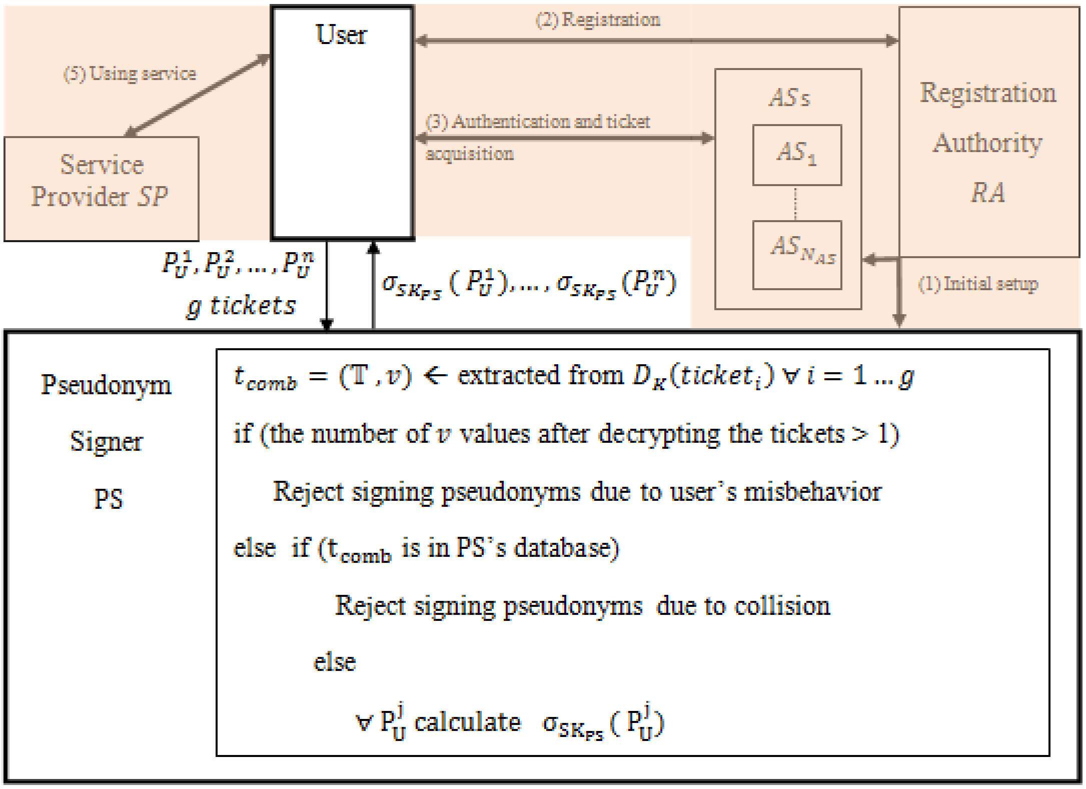
The use of token in this process is mainly to verify the legitimacy of the ticket owner. This verification will be done by $PS$, as detailed in the next section. Moreover, it is worthwhile to note that this verification is done anonymously. Moreover, since the tokens are picked from the token pool randomly and with replacement, a particular token may be used for several users by the same $AS$. This helps to provide unlinkability between the *tickets* and user identities, $ID_U$s, even for the issuing $AS$.

The use of verification code further improves security of CoRPPS by associating the *tickets* to $ID_U$s, $AS$ group identities and session counters. However, this association may work against unlinkability since it may also serve as a unique identifier to link a *ticket* to a user, for the issuing *AS*. Thus, we do not use the hash value entirely, but in modulo $V_{max}$, in order to increase the chance of using a particular *token - verification code* pair for more than one user by an $AS$.

Last, but not the least, since we employ more than one $AS$s in ticket generation process, we enforce collusion of several $AS$s for an attack. In Chapter 5, we provide analyses of these security and anonymity related performance issues.

### 4.5.4 Signing Pseudonyms

In this stage, the user first generates a set of pseudonyms, $P_U^1, P_U^2, \dots, P_U^n$, and sends them to the pseudonym signer, PS, along with $g$ tickets she received from the authentication servers. $PS$ decrypts the tickets to obtain the tokens and the verification codes (for the sake of simplicity of the explanation, we denote the set of these tokens as $\mathbb{T}$ and the combination of $(\mathbb{T}, \mathrm{v})$ as $t_{comb}$). This process is shown in Figure 9.


User
(2) Registration
(5) Using service
Service Provider $SP$
(3) Authentication and ticket acquisition
$AS$s
$AS_1$
$AS_{N_{AS}}$
Registration Authority $RA$
(1) Initial setup
$P_U^1, P_U^2, \dots, P_U^n$
$g$ tickets
$\sigma_{SK_{PS}}(P_U^1), \dots, \sigma_{SK_{PS}}(P_U^n)$
Pseudonym Signer PS

```
$t_{comb} = (\mathbb{T}, v) \leftarrow$ extracted from $D_K(ticket_i)\ \forall i = 1 \dots g$
if (the number of $v$ values after decrypting the tickets > 1)
    Reject signing pseudonyms due to user's misbehavior
else if ($t_{comb}$ is in PS's database)
        Reject signing pseudonyms due to collision
    else
        $\forall P_U^j$ calculate $\sigma_{SK_{PS}}(P_U^j)$
```

Figure 10: Signing pseudonyms

Verification codes in all tickets should be identical for a legitimate pseudonym signing request. $PS$, firstly, makes this equality check. Moreover, $PS$ also checks whether the tokens are legitimate or not. This control is performed by checking whether these tokens exist in the token pool, which the $PS$ obtained in the initial setup phase, or not. If these two tests are successfully passed, then $PS$ checks whether or not it previously issued pseudonyms for another request with the same $t_{comb}$, by searching its database of issued pseudonyms. If there is match, then the same set of tickets and verification code has been either used previously to obtain pseudonyms or there is an incidental collision. CoRPPS does not differentiate between these two cases; but, in either case, $PS$ rejects signing pseudonyms.

At this point, the concept of $t_{comb}$ collision should be detailed. Since the tokens are selected randomly and with replacement from the token pool, and the verification code is calculated modulo $V_{max}$; there is risk of having the same $t_{comb}$ value for two different pseudonym signing requests. Such an incident is called *collision*. If this is the case, then $PS$ declines the pseudonym signing request as described above. Actually, the case of incidental collision causes confusion during the procedure of identity revealing for liability; i.e. the authorities cannot make sure who has used the pseudonyms. Fortunately, with a proper selection of CoRPPS parameters, the probability of collision can be extremely reduced as will be discussed in Chapter 5.

In case of no collision, $PS$ signs all pseudonyms and send the signatures, $(\sigma_{SK_{PS}}(P_U^1), \sigma_{SK_{PS}}(P_U^2), \ldots, \sigma_{SK_{PS}}(P_U^n))$, back to the user. $PS$ records user's pseudonyms $(P_U^1, P_U^2, \ldots, P_U^n)$ associated with $t_{comb}$ to its database. This will enable $PS$ to reveal the *tickets* that are associated with a particular *pseudonym* for identity revealing or revocation purposes.

### 4.5.5 Using the Service

In order to use services, the user sends her pseudonym, $P_U^i$, the signature of pseudonym signer, $PS$, over that pseudonym, $\sigma_{SK_{PS}}(P_U^i)$, and the services she is willing to receive to the service provider, $SP$. $SP$ verifies the signature of $PS$ over the user pseudonym using the public key of $PS$. If the verification succeeds, then $SP$ provides the user with the required services. Figure 10 shows this process.

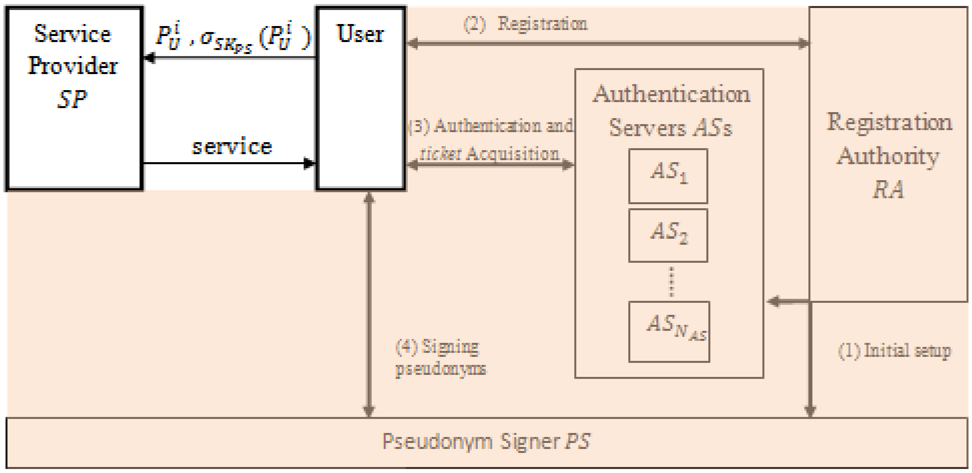

Service Provider *SP*
$P_U^i$ , $\sigma_{SK_{PS}}(P_U^i)$
service
User
(2) Registration
(3) Authentication and *ticket* Acquisition
Authentication Servers *AS*s
$AS_1$
$AS_2$
$AS_{N_{AS}}$
Registration Authority *RA*
(4) Signing pseudonyms
(1) Initial setup
Pseudonym Signer *PS*

Figure 11: Using the service

*SP* maintains a database of services provided and pseudonyms. In each record of this database, the services provided and the pseudonym used to get these services are listed along with date, time and other relevant data

## 4.6 CoRPPS's Features

In this subsection, we describe some extra features supported by CoRPPS. These features stem from the required characteristics and functionality that CoRPPS should provide as a pseudonym providing system. They include flexibility, identity revealing for liability, and pseudonym revocation.

### 4.6.1 Flexibility

By flexibility, we mean the possibility of using CoRPPS as a general anonymous access system for different services. Since service providing systems vary according to the nature of the service, the following CoRPPS parameters can be tuned to fit for a wider range of service providers:

1. Maximum number of pseudonyms allowed to be signed in each session: $\text{NP}_{\text{max}}$
2. The time period that unused signed pseudonyms are valid through: $\text{VT}_{\text{period}}$
3. The lifetime of a pseudonym after its usage: $\text{VF}_{\text{period}}$

The choice of the abovementioned parameters depends on the privacy threats and the required privacy level. If the service relies on information that may lead to privacy leakage such as location information in location based services, then the parameter selection should reflect increased $NP_{max}$ and decreased $VT_{period}$ and $VF_{period}$. If the service does not maintain records of sensitive data, then $NP_{max}$ can be decreased and both $VT_{period}$ and $VF_{period}$ can be increased.

### 4.6.2 Identity Revealing for Liability

One of the main design criteria of CoRPPS is to achieve unlinkability between a pseudonym and the identity of its owner. However, law enforcement units may require to learn the identity of a pseudonym holder in case of a service abuse; a practical system should also support such an identity revealing for liability reasons. The proposed CoRPPS system has the ability to reveal the real identity of a particular user for law enforcement purposes. The process of revealing a real identity is carried out by collaboration among all CoRPPS trusted parties, $RA$, all $AS$s, $PS$ and $SP$; the user entities do not take part in this process. Figure 11 shows the steps of revealing a particular user identity.

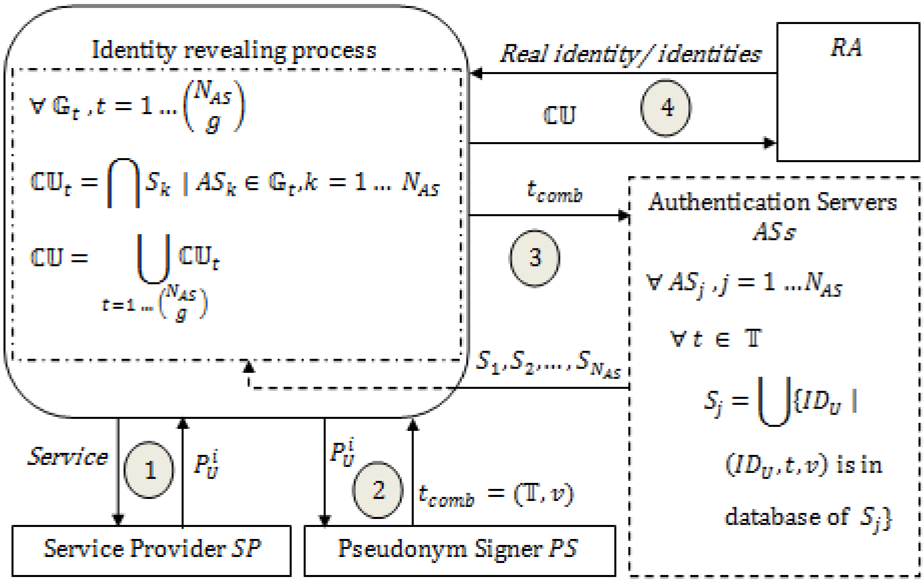

Identity revealing process
$\forall \mathbb{G}_t, t = 1 \ldots \binom{N_{AS}}{g}$
$\mathbb{CU}_t = \bigcap S_k \mid AS_k \in \mathbb{G}_t, k = 1 \ldots N_{AS}$
$\mathbb{CU} = \bigcup_{t=1 \ldots \binom{N_{AS}}{g}} \mathbb{CU}_t$
Real identity/ identities
RA
$\mathbb{CU}$
4
$t_{comb}$
3
Authentication Servers
$ASs$
$\forall AS_j, j = 1 \ldots N_{AS}$
$\forall t \in \mathbb{T}$
$S_j = \bigcup \{ID_U \mid$
$(ID_U, t, v)$ is in
database of $S_j\}$
$S_1, S_2, \ldots, S_{N_{AS}}$
Service
1
$P_U^i$
$P_U^i$
2
$t_{comb} = (\mathbb{T}, v)$
Service Provider $SP$
Pseudonym Signer $PS$

Figure 12: Identity revealing

In step (1), the service, for which the corresponding pseudonym is to be revealed, is sent to service provider, $SP$. $SP$, then, queries its database and returns the target pseudonym, $P_U^i$. In step (2), $P_U^i$ is sent to the pseudonym signer, $PS$, which returns the corresponding combination of tokens and the verification code, $t_{comb}$, by searching its database. In step (3), $t_{comb}$ is sent to all authentication servers, $AS$s, in the system. Each $AS$ queries its database for the set of all user identities, $ID_U s$, to whom any combination of tokens and verification code, $(\mathbb{T}, v)$, was given. The result, $S_j$, of each $AS_j$'s query is sent back to the identity revealing process. Actually, results from the group of $AS$s that took part in generation of $P_U^i$ would suffice, but the pseudonyms, tokens and the verification codes do not carry this information; thus, all $AS$s are needed to be queried. To finish step (3), the identity revealing process takes the intersection $S_j$s for each group of $AS$s (remember that $AS$s are grouped in the setup phase; each group has $g$ $AS$s and there are $\binom{N_{AS}}{g}$ groups). The intersection set for each group, denoted as $\mathbb{CU}_t$, is an empty set if the corresponding $AS$ group has not been employed in the signing process of $P_U^i$.

Finally, the union of all $\mathbb{CU}_t$ sets are calculated to find out the candidate set of $ID_U$s, denoted as $\mathbb{CU}$. The set $\mathbb{CU}$ is, actually, the set of the user *ID*s for which real identities are to be revealed by $RA$. In step (4), $\mathbb{CU}$ is sent to $RA$, which returns the real identity/identities of the user(s) in $\mathbb{CU}$, since $RA$ keeps the $ID_U$ – real identity mappings in its database.

Normally, the set $\mathbb{CU}$ should contain only one user identity ($ID_U$ in our case), if there is no *collision*. However, if there exists a collision, then the number of elements in $\mathbb{CU}$ is greater than one, and consequently, more than one real identity is returned by $RA$ in step (4). Remember that collision occurs if two or more users are incidentally given the same $t_{comb}$ from their corresponding group $AS$s. CoRPPS does not solve the ambiguity caused by collisions, but as we will discuss in Section 4.3, the probability of having a collision can be significantly reduced by selecting the parameters carefully. The reader should notice that that CoRPPS does not sign pseudonyms in presence of collision; the aim of reducing collision probability here is for the sake of network performance, not for security or privacy issues.

Let us give a relatively toy example for the identity revealing process. Assume that we have a CoRPPS system of $N_{AS} = 4$ $AS$s grouped in $g = 3$ $AS$s/group. Therefore, the total number of groups is $\binom{4}{3} = 4$. These groups are the sets of $\mathbb{G}_1 = \{AS_1, AS_2, AS_3\}$, $\mathbb{G}_2 = \{AS_1, AS_2, AS_4\}$, $\mathbb{G}_3 = \{AS_1, AS_3, AS_4\}$, and $\mathbb{G}_4 = \{AS_2, AS_3, AS_4\}$. These groups are assigned to some users and several tickets that contain tokens and verification codes are given to these users by the corresponding $AS$s. The databases of $AS$s at a certain time are shown in Table 1.

Table 1: ASs' tables of the example

| $AS_1$ | | |
|---|---|---|
| $ID_U$ | token | $V$ |
| 1 | $t_1$ | 10 |
| 2 | $t_1$ | 10 |
| 3 | $t_6$ | 20 |
| 4 | $t_1$ | 10 |
| 5 | $t_8$ | 20 |

| $AS_2$ | | |
|---|---|---|
| $ID_U$ | token | $v$ |
| 1 | $t_2$ | 10 |
| 2 | $t_2$ | 10 |
| 3 | $t_2$ | 11 |
| 5 | $t_7$ | 10 |
| 6 | $t_8$ | 20 |

| $AS_3$ | | |
|---|---|---|
| $ID_U$ | token | $V$ |
| 6 | $t_2$ | 10 |
| 1 | $t_3$ | 10 |
| 3 | $t_3$ | 20 |
| 4 | $t_5$ | 10 |
| 5 | $t_3$ | 10 |

| $AS_4$ | | |
|---|---|---|
| $ID_U$ | token | $v$ |
| 6 | $t_3$ | 10 |
| 2 | $t_5$ | 10 |
| 3 | $t_4$ | 20 |
| 3 | $t_5$ | 11 |
| 5 | $t_2$ | 10 |

Now assume that the identity revealing process has carried out steps (1), and (2), and received the value $t_{comb} = (t_1, t_2, t_3, 10)$ from $PS$. To carry out step (3), each $AS$ queries its databases for $ID_U$ values matching any pair of $(t_1, 10), (t_2, 10), (t_3, 10)$. Each $AS$ sends a set $S_j$ as its answer for this query. According to Table 1, $S_1 = \{1,2,4\}$, $S_2 = \{1,2\}$, $S_3 = \{6,1,5\}$, and $S_4 = \{6,5\}$.

After that, the identity revealing process intersects these $S_j$ values, according to the $AS$ groups, in order to calculate the candidate user ID sets, $\mathbb{CU}_t$, for each group. This process yields:

$\mathbb{CU}_1 = S_1 \cap S_2 \cap S_3 = \{1,2,4\} \cap \{1,2\} \cap \{6,1,5\} = \{1\}$

$\mathbb{CU}_2 = S_1 \cap S_2 \cap S_4 = \{1,2,4\} \cap \{1,2\} \cap \{6,5\} = \emptyset$

$\mathbb{CU}_3 = S_1 \cap S_3 \cap S_4 = \{1,2,4\} \cap \{6,1,5\} \cap \{6,5\} = \emptyset$

$\mathbb{CU}_4 = S_2 \cap S_3 \cap S_4 = \{1,2\} \cap \{6,1,5\} \cap \{6,5\} = \emptyset$

The aggregated candidate user identity set, $\mathbb{CU}$, is calculated as the union of these group $\mathbb{CU}_t$ values:$\mathbb{CU} = \mathbb{CU}_1 \cup \mathbb{CU}_2 \cup \mathbb{CU}_3 \cup \mathbb{CU}_4 = \{1\}$, which means that the target $ID_U$ is 1. This value will be sent to $RA$ to get the corresponding real identity.

Here the readers should notice that identity revealing process requires the collaboration of all trusted entities, $SP$,$PS$,all $AS$s and $RA$, of CoRPPS. It is clear that without the help of $SP$, $PS$ and $RA$, we cannot learn necessary pieces of information of the process. However, one may argue that collusion among some $AS$s, not all of them, may suffice since eventually only the $AS$s of the group of $ID_U$ really helps in the process. However, a particular $AS$ cannot know other $AS$s in a particular group; this information is only at $RA$. Thus, collusion among some $AS$s may help to find out $ID_U$ only probabilistically, which is analyzed in Section 4.7.

### 4.6.3 Revocation

Revocation is the process of stopping to provide service to a user. There could be several reasons to revoke a user, which are out of scope of this paper. In this section, we describe how a user is revoked in CoRPPS. To revoke a user, all his signed pseudonyms should be blocked from accessing a service. The process is shown in Figure 12.

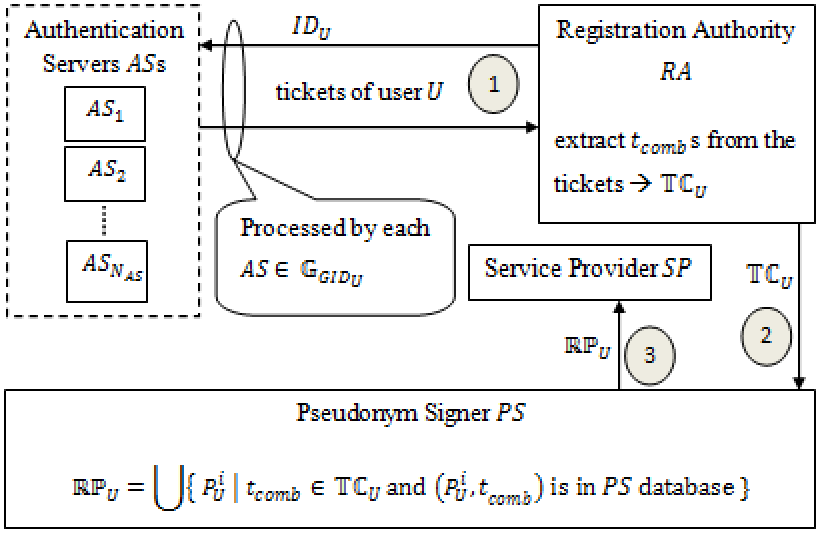

Authentication Servers $AS$s
$AS_1$
$AS_2$
$AS_{N_{AS}}$
$ID_U$
tickets of user $U$
1
Registration Authority $RA$
extract $t_{comb}$ s from the tickets → $\mathbb{TC}_U$
Processed by each $AS \in \mathbb{G}_{GID_U}$
Service Provider $SP$
$\mathbb{TC}_U$
2
$\mathbb{RP}_U$
3
Pseudonym Signer $PS$
$\mathbb{RP}_U = \bigcup \{ P_U^i \mid t_{comb} \in \mathbb{TC}_U$ and $(P_U^i, t_{comb})$ is in $PS$ database $\}$

Figure 13: Revocation in CoRPPS

In step (1), $RA$ sends the identity of the user, $ID_U$, to be revoked to the group of authentication servers she is assigned to. These $AS$s, respond by sending back a set of all tickets issued for $ID_U$. These tickets are listed according to the order of their issuance.

Then, in step (2), $RA$ groups each $g$ tickets of the same order of issuance together, decrypts them, and generates one $t_{comb}$ from each decrypted group of the $g$ tickets. All $t_{comb}$s are then grouped in a set called $\mathbb{TC}_U$. $\mathbb{TC}_U$contains all $t_{comb}$s used by user $U$ for signing pseudonyms. $RA$ then sends $\mathbb{TC}_U$to $PS$ and asks her to find out all *pseudonyms* signed for all $t_{comb}s$ in $\mathbb{TC}_U$. $PS$ lists these pseudonyms in $\mathbb{RP}_U$, which is the set of revoked pseudonyms signed for a particular user $U$.

Finally, in step (3), $PS$ sends $\mathbb{RP}_U$ to the service provider, $SP$. $SP$ updates her revocation list accordingly. Each time a user applies to $SP$ for a service, $SP$ checks the user's pseudonym against the revocation list and then proceeds with the service if the provided pseudonym is not revoked.

### 4.7 Resistance against Attacks

In this section, we describe the level of CoRPPS's resistance against collusion and data disclosure attacks mentioned in Section 4.3.

### 4.7.1 Resistance against Disclosure of Data

The basic idea of CoRPPS design is to prevent the ability of linking a *pseudonym* to a particular user and hence to a particular service. This means that a particular party must not be able to combine both service and identity information. The effects of data disclosure of each entity are listed below.

- $RA$ contains user profiles and real identities, no information about pseudonyms or services can be known by $RA$.
- $AS$ contains user identities and tickets provided to them, no information about pseudonyms or services is available. The number of involved $AS$s in data disclosure affects the level of information they may gain together. This will be discussed in Section 4.
- $PS$ is not able to link the pseudonyms signed for users in different sessions since the tickets contain anonymous but verifiable information. Moreover, $PS$ does not know any information about users or services.
- $SP$ is able to link a particular service to a particular pseudonym; it is not possible for the $SP$ to link a pseudonym to a particular user.

To summarize, in order to find out who has used a particular service, the chain of *pseudonyms*→*tickets*→*User identity*→*Real identity* must be followed and this is not possible without collusion of all trusted entities of CoRPPS. Partial collusions only cause partial problems but do not effectively reveal the real identity of a user who used particular service. A discussion about resistance against collusion among CoRPPS entities is given next. An analytical performance evaluation for the effect of partial collusion is given in Chapter 5.

### 4.7.2 Resistance against Collusions among CoRPPS Entities

Collusion is defined as “a secret agreement between two or more parties for a fraudulent, illegal, or deceitful purpose”. In our case, the attack of collusion among CoRPPS entities, $RA$, $AS$s, $PS$ and $SP$, aims at revealing the real identity of a user who used a particular service. As described in Section 4.6.2, all of these entities must collude together (including all $AS$s) in order to reveal the real identity for liability reasons. On the other hand, it is also possible to have partial collusions, in which some - but not all - of the entities collude. Here, we examine different scenarios of partial collusions between system parties and explain the resistance level of CoRPPS against them.

The collusions between $RA - AS$s, $RA - PS$, $RA - SP$ and $SP - AS$s do not cause any problems since these entity pairs do not have a common information-base to yield the real identity of a user who used a service. Actually collusion among $SP$, $PS$, $AS$s, and $RA$, which is the legal identity revealing process described in Section 4.6.2, is a must for such an attack since the pseudonyms used to access a service is known by $SP$, tickets used to sign a pseudonym is known by $PS$, user identities, $ID_U$s, corresponding to the tickets are known by $AS$s, and real identity of an $ID_U$ is known by $RA$. Although we mention in Section 4.6.2 that all $AS$s must collaborate for guaranteed identity revealing, collusion among a subset of $AS$s may probabilistically suffice as will be discussed now.

Collusion between $SP$ and $PS$ yields the tickets used to obtain a pseudonym, which was used to access a service. Normally a particular $AS$ does not know the other $AS$s in its groups. However, collusion among a subset of $AS$s may cause to identify the groups of $g$ $AS$s that issued the tickets of this pseudonym. This, in turn, causes to identify $ID_U$ and then the real identity with the help of $RA$. As the number of colluding $AS$s increases, the probability of the attack of linking pseudonyms to real identity increases. A detailed analysis of this collusion attack is given in Chapter 5.

One may argue that threshold cryptosystems of $(g, N_{AS})$ described in [76] may be used to enforce authentication with at least $g$ out of $N_{AS}$ authentication servers. In $(g, N_{AS})$ threshold cryptosystems, <u>any</u> subset of $g$ authentication servers out of $N_{AS}$ servers can authenticate a particular user and provide her with the required tickets. However, in CoRPPS we enforce a particular user to communicate only with her corresponding group of $AS$s. This secret mapping makes it harder for authentication servers to collude in the aim of revealing a particular group of users.

Collusion among $SP$, $PS$ and $AS$s causes to run the identity revealing process to some extent such that $ID_U$ who used a particular service can be revealed. Although the real identities cannot be learned in this attack, since $RA$ is not colluding, running this attack several times helps to build a service transaction history for a particular $ID_U$.

CoRPPS allows the $PS$ to sign several pseudonyms in one session, i.e. for each $g$ valid tickets. This feature can be slightly abused by $PS$ by colluding with $SP$. In this way, they can learn that the pseudonyms generated in one session, which belong to one particular user, are used in certain services. Of course neither $PS$ nor $SP$ can link this information to an $ID_U$ or real identity. Moreover linking pseudonyms of different sessions is not possible.

### 4.7.3 *RA-AS-PS* Trio Collusion

A corrupt $RA$ may cooperate with a single $AS$ and the $PS$ to identify all pseudonyms by assigning the corrupt $AS$ to each group during the setup phase. The corrupted $AS$ then replaces $t||v$ in $E_K(t||v)$ with $HMAC(ID_U||Ctr_U)$ truncated or extended to the appropriate length. The $PS$ can then recognize these identifiers and associate a pseudonym with a particular user. Since it is assumed that the encryption scheme is secure, no user will detect this attack.

Fortunately, this attack can be understood by legitimate $AS$s. The total number of $AS$s ($N_{AS}$) and the number of $AS$s per group ($g$) are publicly known. Then it is easy to infer the expected number of groups assigned to each $AS$. Therefore, $AS$s other than corrupted $AS$ in the mentioned attack can easily discover this attack by the significance decrease in the number of groups they are assigned to. For example, if we have a CoRPPS system of $N_{AS} = 10$ $AS$s with $g = 3$ $AS$s per group, then the number of groups a particular $AS$ belongs to equals 72. On the other hand, if we are having a particular $AS$ in each group, the number of groups per each of the other $AS$s is only 8.

A related argument could be to assign users to groups that have a particular $AS$ all the time. However, this time the number of users assigned to the corrupted $AS$ will be much larger than the number of users assigned to legitimate $AS$s. Thus, such an attack can be understood as well.

### 4.7.4 Resistance against Collusions among CoRPPS Users

Another attack is the collusion among two or more users in order to escape from liability. Remember that one of the features of CoRPPS is that real identities can be revealed by law enforcement units for a liability issue. In order to smoothly run this process, a particular user should obtain her tickets from her designated group of $AS$s. In this attack, the cheater user exchanges some tickets with some other users and submits a mixed set of tickets to the $PS$ to obtain signed pseudonyms. In this way, the cheating user seems to obtain tickets from some $AS$s other than her group of designated $AS$s. This situation causes the identity revealing process to fail and, therefore, the cheater cannot be tracked down by law enforcement. However, in order to succeed, the cheater should submit tickets of other users that can be verified by $PS$; this is not so possible, as discussed below.

For two colluding users, with known $ID_U$ and $Ctr_U$, it is not possible to calculate verification code precisely. This is because they do not know the group ID, $GID_U$, which is incorporated in the verification code calculation, $v = hash(ID_U||Ctr_U||GID_U) \bmod V_{max}$. Moreover, they cannot obtain $v$ out of the tickets since the tickets are encrypted and the users do not know the encryption key $K$. However, it is still possible to exchange tickets and to have the same verification code with a probability of $1/V_{max}$, where $V_{\max}$ is the upper bound of the verification code values. As mentioned in Chapter 5, the typical value of $V_{max}$ is 100; this means that the probability of successfully choosing a ticket of the same verification code is only 1%. Moreover, submitting another user's ticket to $PS$ is a blind trial for the cheating user. The reason is that users exchanging the tickets cannot precisely determine that the exchanged tickets have the same verification code, because they do not know $GID_U$.

It is easy to discover such an attack by comparing the verification code of each ticket. Fortunately, it is also possible to identify the cheating users by careful selection of CoRPPS parameters. In a CoRPPS system of $g$ $AS$s in each group, the best chance for a successful attack is to use $g-1$ tickets having the same verification code (i.e. generated for the same $ID_U$, $Ctr_U$, and $GID_U$) and then try one ticket from another user. The $g-1$ tickets alone may then help in revealing the identity of abusive user if we design CoRPPS to have a very low collision probability for $g-1$ $AS$s. The process of identifying abusive users is summarized below:

1. $PS$ detects this attack by testing verification codes and storing ticket combinations involved in each trial.
2. If the number of trials exceeds a threshold, $PS$ reports $RA$ with trials and ticket combinations.
3. $RA$ then runs identity revealing process described in Section 4.6.2 for all combinations of $g-1$ tickets in each trial.

The resulting set of identities contains the abusive users, and due to the very low collision probability, the resulting set will contain very few user identities. Users in the set can then be investigated and abusive ones will be identified and then revoked.

## 4.8 Summary

In this chapter, we explained in details the design and flow of our Collusion Resistant Pseudonym Providing System, CoRPPS. We also explained the properties of CoRPPS and its resistance against attacks. In the next chapter, we will show the performance evaluation of CoRPPS with different metrics.

# 5. PERFORMANCE ANALYSIS FOR CORPPS

We provide detailed performance evaluation of CoRPPS in this section. These analyses cover different issues concerning anonymity and attack resistance.

## 5.1 Anonymity Analysis

Users apply to $AS$s for $tickets$, and each $AS$ maintains records of user's identity and her issued $tickets, (ID_U, ticket)$. A user's $ID_U$ at a particular $AS$ becomes anonymous when that $AS$ issues the same *ticket* for other $ID_U$s. This means that an $ID_U$ is hidden among all other $ID_U$s that have been issued the same $ticket$ in $AS$'s records. In the literature, $k$-anonymity metric is widely used to describe the anonymity level; it refers to the state of being anonymous among other $k-1$ objects [59]. In CoRPPS, a particular $ID_U$ is $k$-anonymous at a particular $AS$ if there exist other $k-1$ $ID_U$s in $AS$'s records with the same $ticket$. For a particular $AS$, the $ID_U$'s anonymity level of that $AS$ is defined as the least number of $ID_U$s that belong to that $AS$ and have been issued the same $ticket$.

To generate a ticket, $AS$ calculates the verification code, $v$, and then picks a *token* , $t$, randomly from the token pool. Both $t$ and $v$ are concatenated and then encrypted by $K$. The total number of distinct $tickets$ is $V_{max} \times N_t$, where $N_t$ is the number of tokens in the token pool, and $V_{max}$ is the upper limit of verification codes, $v$. The probability of reusing a particular ticket is $p = 1/(V_{max} \times N_t)$. The total number of trials performed by users to get a ticket from a particular $AS$ is $n = N_U \times Ctr_{max}$, where $N_U$ is the total number of users and $Ctr_{max}$ is the upper limit of the users' counter.

The ticket reuse is the source of anonymity in each $AS$. If we consider a ticket reuse as a success, we can model ticket generation process as a binomial experiment. The total number of trials $n = N_U \times Ctr_{max}$, and the probability of success $p = 1/(V_{max} \times N_t)$. The resulting binomial distribution is $B(n, p)$.

We simulated the ticket generation process at a particular $AS$, with total number of users $N_U$ = 1000, maximum value of users' counters $Ctr_{max}$=1000, total number of tokens in token pool $N_t$=1000, and upper limit of verification code values $V_{max}$=100. For each *ticket*, we calculated the number of times it was issued, $Freq_{ticket}$. Then we calculated the probability $P(Freq_{ticket}) = Freq_{ticket}/n$, where $n$ is the total number of trials. Figure 13 shows the results of both the simulation experiment and the fitted binomial distribution $B(1000 \times 1000\,, 1/(100 \times 1000))$.

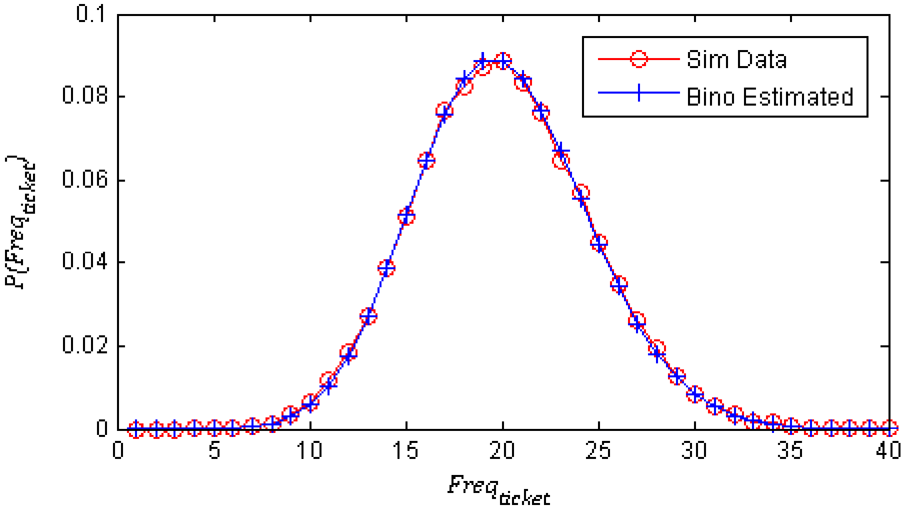


Figure 14: Analytical and simulation results of ticket generation process

As clearly seen in Figure 13, the binomial distribution fits well the ticket generation process. Therefore, the binomial inverse cumulative distribution function[5] can be used to calculate the least number of occurrences of a particular $ticket$, which is the $k$ value of the $k$-anonymity metric. The binomial inverse cumulative distribution returns $k$ with a predefined confidence level, $c$. Table 2 shows $k$-anonymity levels with $c = 0.999$. For $N_U = 10000$ users, and $N_t$=1000 tokens, $k = 71$, which means $k$-anonymity level of 71 is

[5] The Binomial CDF is calculated as $F(k,p,n) = \sum_{i=0}^{k} \binom{n}{i} p^i (1-p)^{n-1}$, where $k$ is the number of successes (in our case anonymity level), $p$ is the probability of success, and $n$ is the number of trials. The inverse of this function returns the $k$ value for a particular probability $c = F(k,p,n)$.

guaranteed with 0.999 confidence level. Table 2 also shows that that the anonymity level is directly proportional to the number of users, $N_U$, and inversely proportional to the number of tokens, $N_t$, in token pool. However, these parameters have a negative effect on the collision probability as will be discussed in Section 5.3.

Table 2: Anonymity level with $\mathrm{c} = 0.999$ , $Ctr_{max}$=1000, $V_{max}$=100

| $\boldsymbol{N_U}$ | $\boldsymbol{k}$ | |
|---|---|---|
| | $\boldsymbol{N_t}$ **=1000** | $\boldsymbol{N_t}$ **=10000** |
| 10000 | 71 | 2 |
| 50000 | 432 | 30 |
| 100000 | 904 | 71 |
| 250000 | 2347 | 203 |
| 500000 | 4783 | 432 |
| 1000000 | 9692 | 904 |

## 5.2 Analysis of Collusion Among *AS*s

$AS$s are grouped in groups of $g$ members of total number of groups $g\_No = \binom{N_{AS}}{g}$, where $N_{AS}$ is the number of $AS$s in the system, and $g$ is the number of $AS$s in each group. Each user is assigned a particular group and should apply only to $AS$s of that group. As a result, if the records stored at $AS$s of a particular group is disclosed, then all *tickets* provided by those $AS$s to users of the same group are disclosed as well. As discussed in Section 5.1, each $AS$ maintains a level of anonymity against the disclosure of its data. This anonymity is the guard of the users' privacy. As the number of colluding $AS$s increases, the probability of revealing the tickets they issued to a particular user also increases. In this subsection, we study the effect of collusion among $AS$s on the disclosure of arbitrary and particular groups, and hence the disclosure of the tickets issued for users belonging to these groups.

### 5.2.1 Collusion among *AS*s for an arbitrary group disclosure

As discussed previously, in CoRPPS, each user is assigned to a particular group of $g$ $AS$s. If $g$ arbitrary $AS$s collude together, then they will be able to reveal all ticket combinations provided for users belonging to the group to which these $AS$s are assigned. This attack does not apply to a particular group; since arbitrary $AS$s collude in this attack, the disclosed group is also arbitrary.

When more than $g$ $AS$s collude, the total number of arbitrarily disclosed groups starts to increase. The number of arbitrarily disclosed groups by collusion among $x$ $AS$s is calculated as $N_{AGD}(x) = \binom{x}{g} = \frac{x!}{g!(x-g)!}$ , where $x \geq g$ is the number of colluding $AS$s, and $g$ is the number of $AS$s per group. Table 3 shows $N_{AGD}(x)$ for a system of $N_{AS} = 10$ $AS$s and $g = 4$ $AS$s in each group. As the number of colluding $AS$s, $x$, increases, the number of disclosed groups, $N_{AGD}(x)$**,** also increases. At least $g$ $AS$s must collude to reveal an arbitrary group. Moreover, to reveal the entire groups, all $AS$s must collude which makes the attack much more difficult. When $x = 5$ (half of the $AS$s are colluding), then only 5 out of the 210 groups are revealed, i.e. only 2.38% of total groups. Even with 70% of all $AS$s are colluding ($x = 7$ in our case), only 16.67% of all groups (35 out of 210) are revealed. This analysis shows that CoRPPS exhibits good resiliency against $AS$ collusion attacks.

Table 3: Number of arbitrarily disclosed groups by collusion among $AS$s

| $\boldsymbol{x}$ | $\boldsymbol{N_{AGD}(x)}$ |
|---|---|
| 4 | 1 |
| 5 | 5 |
| 6 | 15 |
| 7 | 35 |
| 8 | 70 |
| 9 | 126 |
| 10 | 210 |

### 5.2.2 Collusion among ASs for a particular group disclosure

The aim of this attack is to disclose the ticket combinations of the users that belong to a *particular* group. This is a more difficult task for the attacker than the one described in Section 5.2.1 because the attacker needs to know the exact $AS$s of a particular group that a user belongs to and then she has to engage these $AS$s into a collusive agreement. The fact is that the attacker does not have the knowledge-base about the groups to which a particular

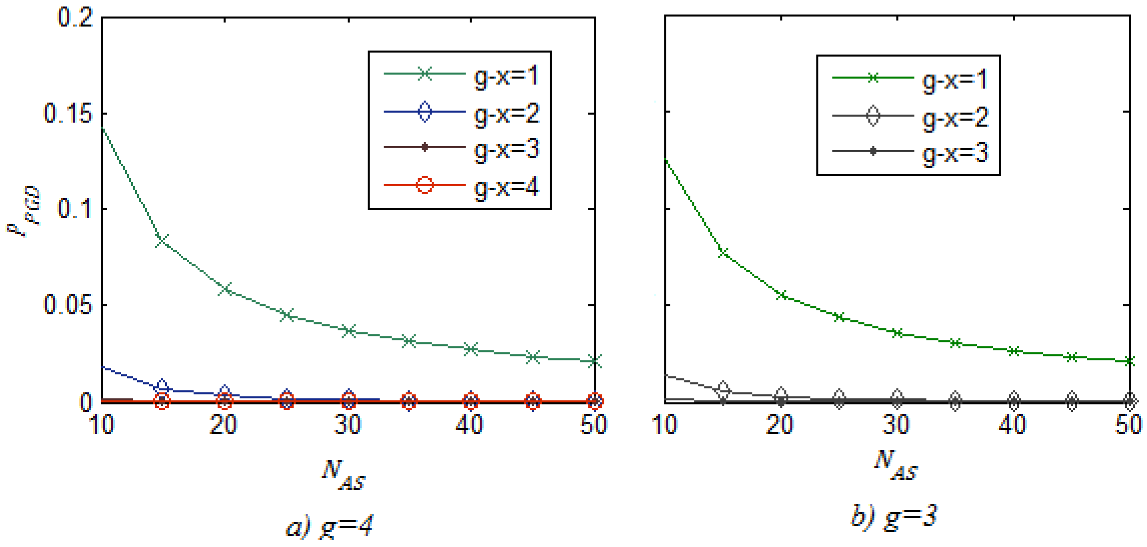
$AS$ belongs without actually engaging it. Therefore, this attack has a probabilistic nature such that the attacker randomly deceives ASs for collusion and then check whether a particular group has been disclosed or not.

Assuming that $x$ represents the number of colluding $AS$s of a particular user's group. The probability of finding the other $(g - x)$ $AS$s of that group, $P_{PGD}(g - x)$, is calculated as $P_{PGD}(g - x) = 1/\prod_{i=x}^{g-x}(N_{AS} - x)$, where $N_{AS}$ is the total number of $AS$s in the system. Table 4 shows $P_{PGD}(g - x)$ for a system of $N_{AS} = 10$ $AS$s and $g = 4$ $AS$s in each group. When none of the $AS$s of a particular user's group are colluding $(g - x = 4)$, the probability of disclosing the entire group of that user is very small (0.00012). Moreover, the probability that a dishonest $AS$ finds other $g - 1$ $AS$s to collude in order to reveal a group is also small (0.00198). If two $AS$s belonging to a user's group have already colluded, then the probability of revealing other two $AS$s, belonging to that group is only 0.01786, which is quite small. The analyses so far show that in the case of partial information about the $AS$s belonging to a particular group, the probability of revealing the entire group is very small. In the worst case of given three already colluded $AS$s, the probability of successfully revealing the other $AS$ is only 0.14286.

Table 4: Probability of revealing a particular group when some $AS$s have already colluded

| $\boldsymbol{g - x}$ | $\boldsymbol{P_{PGD}(g - x)}$ |
|---|---|
| 4 | 0.00012 |
| 3 | 0.00198 |
| 2 | 0.01786 |
| 1 | 0.14286 |

This probability, $P_{PGD}(g - x)$, is inversely proportional to $N_{AS}$ and $g$. This fact is illustrated in Figure 14.



*a) g=4*

*b) g=3*

Figure 15: Effect of increasing the number of $ASs$ on $P_{PGD}$ (a) $g$=4 and (b)g=3

As shown by Figure 14, when  increases, $P_{PGD}$ decreases. Especially when two or more ASs need to collude ($g - x > 1$), $P_{PGD}$ is negligible for $N_{AS} \geq 25$. Increase in $g$ means that more $ASs$ have to collude for compromising the entire targeted group. This provides extra resistance for larger $g$ when the same amount of $AS$s have already colluded, i.e. for same $x$ values. For example, when $x = 2$, $g - x = 2$ for $g = 4$, which corresponds the line with diamond in Figure 14a; and $g - x = 1$ for $g = 3$, which correspınds to line with cross in Figure 14b. When we compare these two lines, we clearly see that larger $g$ has a significant advantage to provide collusion resistance.

### 5.3 Collision Analysis

As mentioned in Section 4.5.4, collision is defined as having the same tokens and verification code, $t_{comb}$ value, for two different pseudonym signing requests. $PS$ should detect collisions and reject signing pseudonyms for colliding requests. As the number of $t_{comb}$s used for signing pseudonyms increases, the collision probability also increases. The collision probability is calculated as $P_{collision} = \frac{EC}{N_{t_{comb}}}$, where $EC$ is the expected number of previously used $t_{comb}$s in signing pseudonyms, and $N_{t_{comb}}$ is the total number of different $t_{comb}$s in the system.

The number of different ticket combinations is calculated as $N_{t_{comb}} = (N_t)^{\,g} \times V_{max}$, where $N_t$ is the number of tickets in ticket pool, $g$ is the number of $AS$s in a group, and $V_{max}$ is the maximum value of the verification code $V$. Each time a user applies to $PS$, $EC$ is incremented by one in the absence of collision, or remains the same if collision occurs. On this basis, $EC$ is defined recursively in terms of the number of times, $n$, different users apply to $PS$ for signing pseudonyms as:

$EC(n) = EC(n-1) + 1 \times \left(P_{no_{collision}}\right)$, By substituting $P_{no\_collision} = 1 - P_{collision} = 1 - \frac{EC(n)}{N_{t_{comb}}}$ we get $EC(n) = EC(n-1) + 1 \times \left(1 - \frac{EC(n)}{N_{t_{comb}}}\right)$. And by putting $x = \left(\frac{N_{t_{comb}}}{N_{t_{comb}}+1}\right)$, we get $EC(n) = x \times (EC(n-1) + 1)$.

The later is a recurrence with basis $EC(1) = 0$. By solving this recurrence using repeated substitutions we get

$EC(n) = x^{n-1} + x^{n-2} + \cdots + x = \sum_{i=1}^{n-1} x^i = \frac{x \times (1 - x^{n-1})}{1-x}$ , and by substituting the value of $x = \left(\frac{N_{t_{comb}}}{N_{t_{comb}}+1}\right)$, we get $EC(n) = N_{t_{comb}} \times (1 - x^{n-1})$, but $P_{collision} = \frac{EC}{N_{t_{comb}}}$ and substituting $EC$ gives us $P_{collision} = (1 - x^{n-1})$

We performed simulations for the pseudonyms signing process, and calculated the collision probability for different values of users' counters, $Ctr_U$. For each $Ctr$ value, we generated the combinations, $t_{comb}$s, and the probability of collision was calculated as the number of colliding requests, i.e. the requests with a previously used $t_{comb}$, divided by the number of non-colliding ones. The collision probability values obtained via simulations and using analytical formulation given above are compared in Figure 15 with the total number of tokens $N_t$=200, maximum value of verification code $V_{max}$=20, total number of users $N_U$=100000, maximum value for users' counters $Ctr_{max}$=1000, and the number of $AS$s in a group $g$=3.

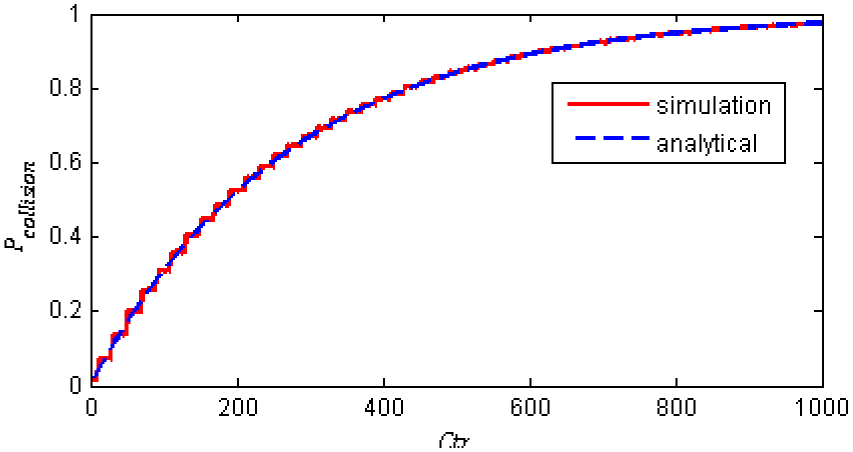


Figure 16: Simulation-based and analytical collision probability ($N_t$=200, $V_{max}$=20, $N_U$=100000, $Ctr_{max}$=1000, and $g$=3)

Figure 15 clearly shows that collision probability fits well our analytical model, $P_{collision}(n) = (1 - x^{n-1})$, where $x = \left(\frac{N_{t_{comb}}}{N_{t_{comb}}+1}\right)$. Table 5 shows the calculated $P_{collision}$ with different values of $N_t$ and $g$, using the analytical model.

Table 5: Collision probability using analytical model, $Ctr$=1000, $V_{max}$=100

| | ***g*=3** | | ***g*=4** | |
|---|---|---|---|---|
| $\boldsymbol{N_U}$ | $N_t$ | | $N_t$ | |
| | **1000** | **10000** | **1000** | **10000** |
| 10000 | $5.98\times10^{-4}$ | $5.99\times10^{-7}$ | $2.39\times10^{-}$ | 0 |
| 50000 | $3.00\times10^{-3}$ | $2.99\times10^{-6}$ | $1.19\times10^{-}$ | 0 |
| 100000 | $6.00\times10^{-3}$ | $5.99\times10^{-6}$ | $2.39\times10^{-}$ | 0 |
| 250000 | $1.48\times10^{-2}$ | $1.49\times10^{-5}$ | $5.98\times10^{-}$ | 0 |
| 500000 | $2.95\times10^{-2}$ | $2.99\times10^{-5}$ | $1.19\times10^{-}$ | 0 |
| 1000000 | $5.81\times10^{-2}$ | $5.99\times10^{-5}$ | $2.39\times10^{-}$ | 0 |

From Table 5, we see that collision probability can be reduced tremendously with proper selection of parameters. As the number of users, $N_U$, increases, the probability of collision also increases. Moreover, as the number of tokens, $N_t$, in token pool increases, the probability of collision decreases significantly. In Section 5.1, we showed that as the number of tokens, $N_t$, in a key pool increases, the anonymity level at a particular $AS$

decreases. Therefore, the choice of $N_t$ becomes a tradeoff between anonymity level and collision probability. Fortunately, it is possible to maintain the same level of anonymity while decreasing the collision probability by increasing the number of $AS$s in a group, $g$. As $g$ increases, the probability of collision decreases.

## 5.4 Communication Complexity in CoRPPS

To sign pseudonyms, a user has first to acquire $g$ tickets from his assigned authentication servers. The she sends her generated pseudonyms with the acquired $g$ tickets to the pseudonym signer for the purpose of signing them. In this section, we formulate the communication complexity for ticket acquisition and for pseudonym signing. Then, we will analyse this cost for a particular CoRPPS parameters. It is worth mentioning here that ticket acquisition and signing pseudonyms can be done offline.

### 5.4.1 Formulation of Complexity for Tickets Acquisition

To acquire tickets, a user runs authentication and ticket generation process described in Section 4.5.3 $g$ times, where $g$ is the number of $AS$s in each group. The length of the user's identity, $ID_U$, depends on the maximum number of users in the system $MAX_U$ and equals $log(MAX_U)$ [6], similarly, the size of the counter is equal to $log(Ctr_{max})$. Thus the size of the data sent by a user to a particular authentication server equals $log(MAX_U) + log(Ctr_{max})$. The user will then receive a random challenge of size $L_{RC}$ and will respond to the challenge with user response of size $L_{UR}$.

A ticket is composed of a token and a verification code as described in Section 4.4.4. The length of the token depends on the size of the token pool, $N_{t_{comb}}$ and equals to $log\ (N_{t_{comb}})$. Similarly, the length of the verification code is $log(V_{max})$, where $V_{max}$ is the

[6] Logarithms here are calculated to the base 2.

maximum possible value of the verification code $v$. Thus, the entire length of each ticket is $L_t = log\ (N_{t_{comb}}) + log(V_{max})$.

The size of data sent and received by a user for a ticket acquisition from a particular authentication server equals $TA_{size} = L_{RC} + L_{UR} + log(MAX_U) + log(Ctr_{max}) + L_t$. Since we have $g$ authentication servers, then we will have $TA_L = g \times (L_{RC} + L_{UR} + log(MAX_U) + log(Ctr_{max}) + L_t)$ bits of traffic for the entire ticket acquisition process.

### 5.4.2 Formulation of Complexity for Signing Pseudonyms only

In CoRPPS, pseudonyms are public keys that are signed in batches of at most $NP_{max}$ pseudonyms. The size of a pseudonym depends on the underlying algorithm used for the generation of these pseudonyms. Denoting the length of a pseudonym by $L_P$, then the user will have to send $L_P \times NP_{max}$ bits for the pseudonyms and $g \times L_t$ bits for the tickets.

Then the user will receive the signatures of pseudonym signer over these pseudonyms. The size of the signature depends on the size of the key of the algorithm used in signing. Assuming that $PS$ uses the same key length used by users to generate their pseudonyms, then the size of the signature becomes the same as the size of the pseudonym. Thus, the total number of bits for signing $NP_{max}$ pseudonyms is equal to $PS_L = 2 \times L_P \times NP_{max} + g \times L_t$.

### 5.4.3 Communication Complexity Analysis

According to Sections 5.4.1 and 5.4.2, the size of traffic required for acquisition of tickets and signing pseudonyms is equal to $TA_L + PS_L$. Assume that we have a CoRPPS system with typical values of $N_{t_{comb}} = 1024$, $g = 3$, $MAX_U = 2^{20}$, $V_{max} = 128$, and $Ctr_{max} = 1024$. Assume also we use RSA algorithm of 1024 bit key size, i.e. $L_P = 1024$ bits. And assume that we are using SHA1 hash algorithm of 160 bit hash length, then $L_{RC} = L_{UR} = 160$ bits.

Figure 16 shows the size of entire traffic required for ticket acquisition and signing pseudonyms for different number of pseudonym, $NP_{max}$, in each trial.

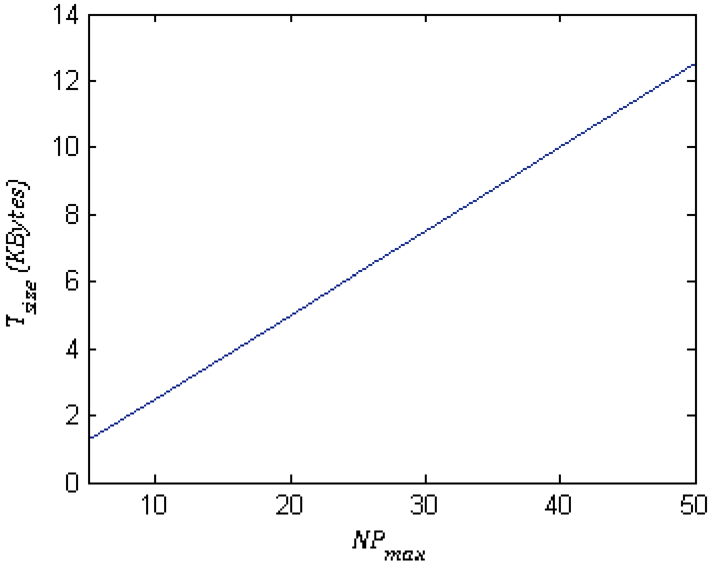


Figure 17: Size of traffic vs. number of signed pseudonyms

As seen from Figure 16, the size of traffic is linear with respect to the number of signed pseudonyms. When the user needs to sign 20 pseudonyms, then the traffic size is about 5.14 kilobytes. For 50 pseudonyms, the user requires 12.64 kilobytes which is an acceptable amount of traffic.

### 5.5 Summary

In this chapter, we provided the performance evaluation of our collusion resistant pseudonym providing system, CoRPPS. Evaluation was based on different metrics including anonymity, collusion resistance, and collision probability. CoRPPS performed well under these metrics and hence we will use it as the pseudonym signer for our privacy aware collaborative traffic monitoring system, PA-CTM that will be explained in the next chapter.

# 6. PRIVACY AWARE COLLABORATIVE TRAFFIC MONITORING SYSTEM USING AUTONOMOUS LOCATION UPDATE MECHANISM

Collaborative Traffic Monitoring (CTM) systems exploit the location information continuously collected from vehicles. Location data is very sensitive information that made privacy a major obstacle for the widespread usage of CTM systems. The way how this data is generated and used is very important for users' privacy and data quality as well. Recently, two CTM approaches have been proposed, the first relies on a dedicated infrastructure which is called Vehicular Ad hoc Networks (VANETs), and the second utilizes the existing underlying infrastructure such as cellular and wireless networks. In this chapter, we propose our Privacy Aware Collaborative Traffic Monitoring System (PA-CTM) that considers the privacy and security properties of VANETs and existing infrastructures. PA-CTM provides a client server architecture that relies on existing infrastructures and enhances privacy firstly by using a robust pseudonym providing system for anonymous access called CoRPPS, which was detailed in Chapter 4. And secondly by utilizing a novel Autonomous Location Update Mechanism (ALUM) that does not rely on a Trusted Third Party and uses only local parameters (speed and direction) for triggering a location update or pseudonym change. Our performance results showed that ALUM is effective for traffic monitoring in terms of both privacy and utility.

## 6.1 Introduction

Traffic monitoring systems have evolved rapidly in the last years due to the advances in communication technologies such as GPS, GSM and 3G networks. The main idea behind Collaborative Traffic Monitoring (CTM) systems is that users provide their location information to a server, in return they can benefit from the system such as viewing current traffic status in a particular region. CTM systems are critical nowadays especially in big cities with heavy and sometimes unpredictable traffic. Widespread usage of CTM systems would alleviate the congestion by proposing alternative routes to users and hence avoid more vehicles entering the congested areas. In this way, CTM systems would save time and money, and more importantly decrease carbon emission by optimizing the traffic [68].

The provision of the user's exact location leads to privacy leakage problems because users may be identified using their locations. Moreover, they may be profiled according to their favorite locations; hence, they may be a target of different spam messages. Despite the fact that these systems use anonymous identities for users, they may still be traced and identified using some data mining techniques [34].

CTM systems are divided into two major disciplines according to the underlying technology. The first one is based on the usage of Dedicated Short Range Communication (DSRC) that supports both Vehicle to Vehicle (V2V) and Vehicle to Infrastructure (V2I) communications [28]. We name these as Dedicated Infrastructure (DI) Systems. The second discipline is the utilization of existing underlying infrastructure such as cellular and wireless networks to set up an Intelligent Transportation System (ITS). The architecture is a client server architecture where clients send their location information to a server and get a complete overview of traffic from the server. We name these systems as Existing Infrastructure (EI) systems.

DI approach requires deployment of dedicated infrastructure that is costly and needs time as well. On the other hand, EI approach utilizes existing communication infrastructure, and does not require new deployments, however it does not support anonymous access using pseudonyms. In EI approach, location privacy preserving techniques depend on a trusted third party and degrades the data quality as well. We develop a privacy aware collaborative traffic monitoring system, PA-CTM, that is based on EI and allow anonymous access using pseudonyms. Our PA-CTM relies on local parameters – NOT on a trusted third party, TTP - for triggering location updates and changing pseudonyms. PA-CTM also reduces communication cost compared to other periodic location updates used in EI systems.

## 6.2 Background

Here we discuss the requirements of a location update mechanism. Then we present the moving object model that we will use in our notations.

### 6.2.1 Requirements of a Location Update Mechanism

From the flaws of existing location update mechanisms described in Section 2.4, we find that any update mechanism should:

- Catch all traffic irregularities that may appear
- Be non periodic
- Not depend on a Trusted Third Party (TTP).

The third condition implies that moving objects should decide by themselves whether to update or not without communicating with other moving objects or parties. Fortunately, we can achieve such a method by utilizing the existence of traffic laws and regulations that drivers are enforced to follow for the purpose of a safe driving.

### 6.2.2 Moving Object Data Model

Sistla et al [69] proposed a data model for representing moving objects in database management systems called Moving Objects Spatio-Temporal (MOST) data model. According to MOST, a moving object $A$ is modeled by three attributes, $A.value$, $A.updatetime$ and $A.function$. The $value$ attribute represents the current position of object $A$, $updatetime$ represents the time at which $A.value$ was updated, and finally $function$ (speed) is a function of time that describes the future $value$ of object $A$, i.e. $A.function$ is used to calculate future $A.value$s. If $A.value(t_0) = value_0$ and $A.function(t_0) = function(t_0)$ at $A.updatetime = t_0$, then $A.value(t_0 + \Delta t)$ is calculated as $A.value(t + \Delta t) = A.value(t_0) + A.function(t_0) * \Delta t$ . i.e. *A.function* is used to calculate the expected location of object $A$ using its last observed $A.function$ attribute and the elapsed time. Note that $A.function$ represents the speed of a moving object when we model traffic. The use of $A.function$ instead of speed is for generalization purposes.

### 6.2.3 Changing pseudonyms

To enhance privacy in CTM systems, users use temporary identities called pseudonyms instead of using their real identities [30]. The use of pseudonyms enables hiding the real identity of users. However, it has been showed that even when using pseudonyms, it may be possible to reveal the real identity of some users by their location information [70].

To overcome this flaw, users are asked to change their pseudonyms from time to time. Changing pseudonyms makes it difficult to link pseudonyms together in the aim to build partial or even complete trajectory of the moving object. This in turn enhances the privacy level of users. However, pseudonym change is of no use if very few users do this change. Few users means higher probability of linking corresponding pseudonyms successfully.

A proposed solution called mix zone has been provided [29,31]. Mix zones are regions where many users are expected to exist such as traffic lights or road intersections. Users are asked to change their pseudonyms inside these Mix zones so that new pseudonyms are mixed together. This will increase the privacy level and will make it difficult for an adversary to link two successive pseudonyms correctly.

## 6.3 PA-CTM architecture and flow

PA-CTM utilizes a client server architecture; users first sign their pseudonyms and then use the signed pseudonyms to authenticate to the traffic server for either updating their location information or querying current traffic status. The traffic server collects location information from different users (identified by pseudonyms) and provides users with current traffic status. Figure 17 shows the general architecture of our PA-CTM system.

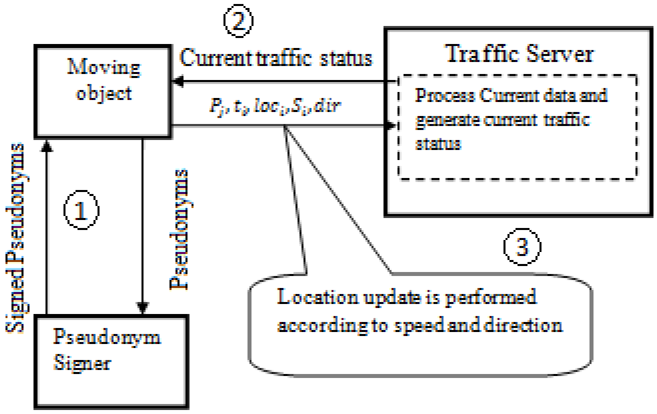

2
Current traffic status
Traffic Server
Moving object
$P_j, t_i, loc_i, S_i, dir$
Process Current data and generate current traffic status
Signed Pseudonyms
1
Pseudonyms
3
Location update is performed according to speed and direction
Pseudonym Signer

Figure 18: PA-CTM architecture and flow

The flow of our PA-CTM is shown on Figure 17. Firstly, step 1, the moving object generates a number of temporary identities called pseudonyms, and then she sends these pseudonyms to the pseudonym signer who in turn signs them and returns the signatures to the moving object. Then, step 2, the moving object can be authenticated to the traffic server using one of her signed pseudonyms. Note that users are capable to change pseudonyms from time to time and hence to divide their real trajectory into smaller trajectories identified by their pseudonyms. This in turn makes it hard to link different pseudonyms in the aim of constructing the complete trajectory. In step 3, the traffic server verifies the signature of the Pseudonyms Signer and responds to the moving object accordingly.

The Pseudonym Signer is responsible for registering users and signing their pseudonyms. Our PA-CTM adopts a pseudonym providing system called Collusion Resistant Pseudonym Providing System (CoRPPS) that was detailed in Chapter 4 [15].

### 6.4 Autonomous Location Update Mechanism, ALUM, Design

When analyzing traffic flow, we can easily expect driver's behavior, and hence the moving object's behavior, according to traffic conditions. Moving objects tend to behave similar under similar traffic conditions, thanks to traffic regulation laws. If drivers face an obstacle they immediately reduce their speed and may be direction to avoid that obstacle. The same is applied when they face congestion; they tend to avoid that congestion by changing their direction to another path. This unified behavior is illustrated in Figure 18 which shows a part of a two sided road with lanes T1 and T2.

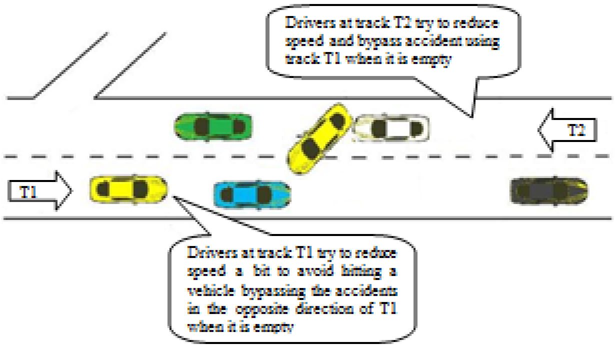

Drivers at track T2 try to reduce speed and bypass accident using track T1 when it is empty
T2
T1
Drivers at track T1 try to reduce speed a bit to avoid hitting a vehicle bypassing the accidents in the opposite direction of T1 when it is empty

Figure 19: Illustration of unified drivers' behavior

In Figure 18, An accident occurred at T2 that caused a block of T2's track. Moving objects through T2 track will reduce their speed and try to pass by T1 when they got an opportunity to do that. At the same time, moving objects at T1 will consider the accident and reduce their speed to bypass the accident region safely even though their track, T1, is clear. This unified behavior of all drivers can be utilized to investigate traffic conditions at different regions.

ALUM relies on the fact that under similar traffic conditions, moving objects will behave similar. Hence ALUM triggers location update and pseudonym change if a change of a particular moving object's behavior occurs. It assumes that all moving object at that region will also update their locations and change their pseudonyms because they will face the same traffic condition, and hence they are expected to behave similarly. Given that a moving object $A$ has two successive location updates at $A.value_1$ and $A.value_2$. Then these two updates must achieve

1. $|A.function_1 - A.function_2| \geq SD_{Thresh}$ , or
2. $A.direction_1 \neq A.direction_2$

We adopted the notations of Sistla et al. in [69], $A.function$ reflects the speed of object $A$ at a particular time stamp. In 1, if the difference between the ($A.function_1$ , $A.function_2$) is greater than a predefined threshold, $SD_{Thresh}$ , then an update and a pseudonyms change are required. This means that $A$ has changed its speed dramatically, in other words the acceleration or deceleration (both terms are used interchangeably) is high. The selection of the best threshold value is essential for ALUM to work efficiently, the details on the choice of $SD_{Thresh}$ are provided in Section 7.2.

Condition 2 examines the change in the direction of $A$ and triggers an update and a pseudonym change if this value has changed to any of the other 7 direction values (namely, SS, EE, WW, NN, SE, SW, NE, and NW). Direction values are relaxed into 8 values as shown in Figure 19. The relaxation is important to avoid frequent changes of directions for changing tracks or bypassing other moving objects.

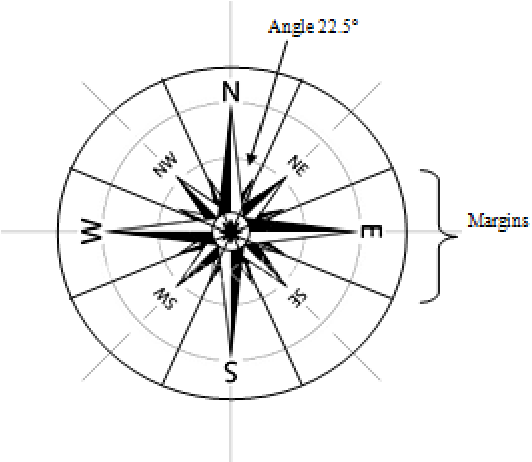


Figure 20: Direction relaxed values

If 1 and 2 are not met, the moving object is assumed not to update. However this non-update period may last for long. To avoid this, moving objects are triggered to update their location and optionally change their pseudonyms after a period of time randomly generated from a predefined interval [0:$T_{Rand}$], where $T_{Rand}$ is the maximum time allowed to wait before the next location update occurs. If conditions 1 and 2 are not met, then the moving object $A$ picks a random value $t$ from an interval of [0, $T_{Rand}$] and waits for time $t$ to update her location. Note that $A$ may update her location and change her pseudonym before $t$ is consumed if conditions 1 or 2 are met again.

Unlike virtual trip lines, VTL, described in [25], ALUM does not need a trusted third party to decide when to update. And by using a random period to update, ALUM does not maintain a regular update pattern which makes linking of location updates more difficult. Because of its ability to capture changes in moving objects' behaviors, ALUM ensures capturing all non regular traffic conditions. This strengthens ALUM against random silent period's mechanism.

ALUM is expected to reduce the communication cost due to the reduced number of required updates compared with periodical update mechanism. More details about communication costs are given in Section 7.6.3.

Under regular traffic conditions, users update their locations according to silent period mechanism. They are also free to change their pseudonyms in these cases too. This will enable users to adjust their required privacy level against traffic server according to their preferences.

## 6.5 Enhanced Autonomous Location Update Mechanism, EALUM

In ALUM, if no significant speed variation or direction change occurs then location update is done after a random period of time. This update may affect privacy level if it was performed in regions of a very low traffic activity. Different regions have different traffic weights according to their traffic activity. For traffic monitoring and privacy concerns, these regions should not be treated equally; the higher the weight of a region the greater the effect of that region on traffic monitoring and on privacy as well.

Privacy in ALUM can be enhanced by considering traffic weights. When ALUM is to update according to the silent period condition, it first checks the weight of her sub region. If the weight is greater than a predefined weight, $W_{Thresh}$, then update is performed. Otherwise, update is delayed for another silent period. We call this mechanism Enhanced ALUM, or shortly EALUM.

According to Hoh et al [24], determining the weights of sub regions can be done by dividing the total region into sub regions. Traffic is monitored for a relatively long period of time over that region. The weight of each sub region is then calculated as the number of updates generated in that sub region divided by the total number of location updates performed in the total region; the higher the weight of a sub region, the larger the probability of having more vehicles in vicinity. This will positively affect the *k*-anonymity level and will reduce the linking probability of two pseudonyms as well.

## 6.6 Properties of ALUM and EALUM

ALUM enhances privacy of users by enforcing them to change their pseudonyms upon speed or direction change, which in turn forms a mix zone. Changing of pseudonyms in mix zones makes it difficult to link two pseudonyms of the same user. Applying ALUM or EALUM also increases the anonymity level by enforcing all users in vicinity to update their locations.

ALUM and EALUM does not update location periodically, hence the data quality will be less than that of periodic update mechanism. But for traffic monitoring purposes, many periodic updates carry the same information and do not enhance traffic monitoring. This explains the capability of ALUM data to fit very well for traffic monitoring purposes as will be shown in Chapter 7.

ALUM and EALUM are expected to reduce the communication cost due to the reduced number of required updates compared with periodical update mechanism. This in turn will increase the users' turnout to PA-CTM due to lower communication costs.

## 6.7 Privacy and Location Prediction Accuracy

Privacy in location based service refers to the state of protecting users' information from being disclosed. It also includes the information gained by applying data mining tools on location information. Possible data mining attacks include users' profiling and tracking [72]. In this section we describe how can location data be privacy invasive, we also present an error model that is assumed to enhance privacy.

### 6.7.1 Privacy invasion via location prediction

Location data points contain sensitive information that may lead to partial or complete disclosure of the user's trajectory that may lead to the disclosure of the real identity of that user. Two or more location updates can be linked using the speed at the previous location and the expected travel time. To calculate the next expected moving object location, $A.value(t+\Delta t)$, from current object location, $A.value(t)$, we use $A.value(t+\Delta t) = A.value(t) + A.function(t) \times \Delta t.$ The precision of the result depends on how far this result from the actual location, we will use the term prediction accuracy to refer to the difference between expected and actual location.

To exactly link two location updates together, one should be aware of *prediction accuracy*. Low *prediction accuracy* means greater difference between actual and expected location, and hence low precision about the exact next location. Because of *prediction accuracy* an adversary needs to try different linking possibilities of all other location updates that occur within the range of *prediction accuracy*, the area that covers this range is called Uncertainties Region, $UR$. As the *prediction accuracy* decreases, $UR$ increases, and hence the probability of having more location updates inside $UR$ increases. This in turn is expected to enhance the privacy level of moving objects.

There are different factors affecting *prediction accuracy*, they stem from different sources of errors in calculating the next expected location. These factors are combined together to form an error model that can be used for linking location updates. This model is detailed in Section 6.7.2.

### 6.7.2 Error Model

As described in Section 6.7.1, the success of linking two successive location updates depends on the *prediction accuracy*. The latter is affected by different errors due to GPS accuracy, $function$ (speed) variation between two successive updates, and propagating errors from previous predictions. In this section, we will detail these errors and show how to use them in calculating uncertainty region, $UR$.

#### 6.7.2.1 GPS Precision error

GPS precision error, $GPS_{ERR}$, is the error associated with the precision of the GPS system and devices used. This error varies from one device to another, modern GPS devices can give a precision error of 14 meters [73]. However, this precision is not maintained all the time, and GPS devices vendors supply extra information for buyers such as the percentage at which this value of precision error is guaranteed. For this reason a vehicle can be positioned in 2 dimensions by a circle of radius $GPS_{ERR}$ , or in 3 dimensions by a sphere of the same radius. This error affects the *prediction accuracy* and increases the radius of $UR$ by the value $GPS_{ERR}$.

#### 6.7.2.2 Location prediction error

To predict the next location of a vehicle from the current location information we use $A.value(t + \vartriangle t) = A.value(t) + A.function(t) \times\vartriangle t$, where $A.value(t + \vartriangle)$ is the predicted location of object $A$ at time $t + \vartriangle t$, $A.value(t)$ is the current location of $A$, and $A.function$ is the speed of $A$ at time $t$. During the time period $\vartriangle t$, $A.function$ value may vary due to traffic conditions, this variation will cause incorrect calculations of $A.value(t + \vartriangle t)$. We will call this type of error location prediction error, $LocPred_{ERR}$.

The prediction model assumes fixed speed over the time period $\vartriangle t$ which is not true because, according to ALUM, an object $A$ may change $A.function$ without updating the database as far as $A.function$ variation does not exceed a threshold, $SD_{Thresh}$. As $\vartriangle t$ increases, we expect $LocPred_{ERR}$ to increase; this is because we will have more speed ($A.function$) variations during this period of time. $LocPred_{ERR}$ depends on the traffic conditions at the region where a moving object moves and on the value $\vartriangle t$ between two successive updates. $LocPred_{ERR}$ affects *prediction accuracy* and increases the radius of Uncertainties Region, $UR$, by $LocPred_{ERR}$.

The value $LocPred_{ERR}$ cannot be directly calculated, it depends mainly on traffic conditions where a moving object moves. It also depends on the time interval $\vartriangle t$ between two successive updates. The larger the time interval $\vartriangle t$, the larger $LocPred_{ERR}$, and hence the larger the radius of $UR$, $R_{UR}$. Calculating the exact $LocPred_{ERR}$ is not an easy task. It requires the knowledge of the exact next location of a moving object which in real world is unknown and needs to be predicted. However, it is possible to obtain an empirical model for this error using historical data and then using this model to estimate $LocPred_{ERR}$. Details of calculating $LocPred_{ERR}$ are shown in Section 7.4.

#### 6.7.2.3 Inherent error

This error occurs when we have more than one moving object inside $UR$ at time $t$, among which is the target object that an adversary is trying to link location updates with those at time $t + \Delta t$. Due to the fact of having more than one moving object in $UR$, we will have more linking possibilities between current and expected $UR$s, i.e. $UR$ at $t$ and $UR$ at $t + \Delta t$ in Figure 20. Expected $UR$ should be wide enough to contain all possible pairings of different pseudonyms. This causes an increase in the radius of the expected $UR$ by the value of the radius of the minimum circle that contains all moving objects in current $UR$. We call this increase as $Inh_{ERR}$. Figure 20 shows the concepts error model.

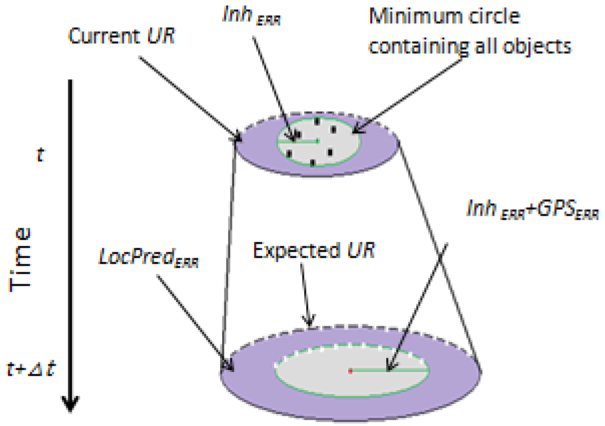


Figure 21: The concept of error model

The strength of $Inh_{ERR}$ to enlarge the expected $UR$ depends on the number of objects in the current $UR$ and how far they are from each other. As the number of objects in current $UR$ increases, $Inh_{ERR}$ increases causing an increase in the expected $UR$. This in turn increases the probability of having more objects in expected $UR$ and hence increasing the uncertainty of linking location updates. If the number of objects in current $UR$ is 1, then $Inh_{ERR}$ will be 0 meaning that it has no effect on increasing expected $UR$. Section 6.8 details how do we empirically calculate $Inh_{ERR}$ from historical moving objects database.

#### 6.7.2.4 Complete Error Model

After introducing the previous error terms, we finalize our error model for calculating expected $UR$ as follows:

If number of moving objects at current $UR$ is only 1 then $Inh_{ERR}$ will be 0, and the expected $UR$ radius will be $GPS_{ERR} + LocPred_{ERR}$
If number of moving objects at current $UR$ is greater than 1 then $Inh_{ERR}$ will be minimum radius of the circle containing all vehicles, and the expected $UR$ radius will be $Inh_{ERR} + GPS_{ERR} + LocPred_{ERR}$.

We will use this error model to calculate the expected $UR$, and then to calculate anonymity level inside it.

### 6.8 *k*-Anonymity Level Calculation

Anonymity can be defined as the state of a moving object $A$ to be hidden among other objects in terms of time and location attributes. i.e. an object is anonymous if there exists other objects with the same values. Because of GPS precision error and other sources of errors, the definition can be relaxed to being hidden among other moving objects in a particular uncertainty region, $UR$. $k$-anonymity metric is widely used to describe the anonymity level, it refers to the state of being anonymous among another $k-1$ objects [59]. We will use this metric to calculate anonymity level at each $UR$ for each point on the trajectory of a moving object. The results of this metric will be used to compare between our update mechanism ALUM and periodical update mechanism. Algorithm 1 details how to calculate the anonymity level.

**Algorithm 1**: Calculating $k - anonymity$ for each location point at a particular vehicle's trajectory

**Input**: $D$: dataset of trajectories identified by pseudonym .

**Output**: $k\_D$: data set of location updates and their $k - anonymity$ values

**Procedure:**

```
1.  $Inh_{ERR} = 0$
2.  for all Pseudonyms $P_j$ $in$ D
3.      for all time stamps $t_i$ where $t_i$ in $\cup P_j.updatetime$
4.          $R_{UR} = GPS_{ERR} + Inh_{ERR} + LocPred_{ERR}$
5.          $E_{P_j.value}(t_i + \triangle t) = A.function(t_i) * \triangle t$
          O = RETRIEVE o WHERE ($o.updatetime = t_i$ AND (|o.$location - E_{P_j.value}(t_i + \triangle t)| \leq_{UR}$) AND ( o.$direction = P_j.direction$) )
6.          $k_{P_j,t_i} = \mathrm{COUNT(O)}$
7.          if( $k_{P_j,t_i} > 1$ )
8.              $Inh_{ERR}$ = radius of minimum circle containing all objects in O.
9.          else
10.             $Inh_{ERR} = 0$
11.     end
12. end
```

The algorithm calculates the $k$-anonymity level for each point on the moving object's (a moving object here is identified by his used pseudonym $P_j$) trajectory. For each $P_j.updatetime$ in object $P$'s trajectory, the algorithm first calculates the radius of uncertainty region, $R_{UR}$, line 4. It then calculates the expected next location associated with pseudonym $P_j$, $E_{P_j.value}(t_i + \triangle t)$. This value is determined by the expected time to the next update, $\triangle t$. Line 6 of Algorithm 1 represents a Future Temporal Logic, FTL, query [69] that retrieves all objects $O$ that have the same update time, same direction, and exist in $UR$. The anonymity level $k$ is defined as the number of these objects in $UR$. If $k > 1$, then $Inh_{ERR}$ should be set as the radius of the minimum circle containing all objects in $UR$, line

9. This new value of $Inh_{ERR}$ will be used in line 4 to update the radius of the expected $UR$, $R_{UR}$, for the next iteration, i.e. next time stamp.

For a pseudonym $P_j$, $P_j.location$ is determined by two values $x$-coordinate and $y$-coordinate respectively. To calculate the radius of the minimum circle containing all moving objects that resides in a particular $UR$, we keep track of the maximum and minimum $x$ and $y$ coordinates of the location updates for these objects, namely $min_x$, $min_y$, $max_x$, and $max_y$. In the Cartesian plane, ($min_x$, $min_y$), ($min_x$, $max_y$) , ($max_x$, $min_y$), and ($max_x$, $max_y$) points can be used to represent a bounding rectangle around all the moving objects inside $UR$. Figure 21 shows this rectangle, $ABCD$. The minimum circle is centered at the midpoint of diagonal $BD$, and called $x$ on the figure. The radius of the minimum circle is $Inh_{ERR} = ||BD||/2$, where $||BD||$the distance between points $B$ and $D$ of the rectangle $ABCD$.

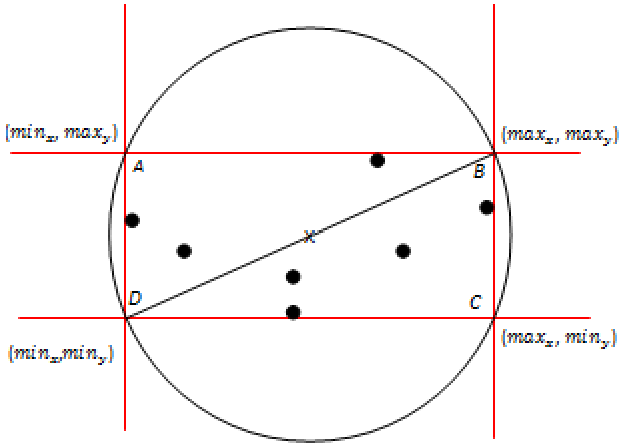


Figure 22: Calculating the radius of the minimum circle

## 6.9 Summary

In this chapter, we explained in details the architecture of PA-CTM. We then detailed the design of ALUM and EALUM as a location update and pseudonym change mechanism. Then we provided an error model used to calculate the radius of $UR$ which was used to calculate the anonymity level $k$. In the next chapter, we will provide the performance evaluation for ALUM and EALUM.

# 7. PERFORMANCE ANALYSIS FOR PA-CTM

In this chapter, we show our experimental results. We explain how to calculate the speed threshold, $SD_{Thresh}$ ,the sub region weight threshold $W_{Thresh}$, and the location prediction error, $LocPred_{ERR}$. A comparison among ALUM, EALUM, periodic and silent period update mechanisms in terms of anonymity and utility are also reported.

## 7.1 Experimental Setup and Dataset

Due to the lack of real GPS datasets, we used Thomas Brinkhoff simulator to generate GPS traces [74]. The generator follows the benchmarks of generating datasets by using real network models with speed and capacity limits. This enabled us to generate a more realistic dataset compared with datasets generated from other generators. The generator allows simulating different traffic scenarios by choosing proper parameter values. Figure 22 shows a map of San Francisco generated by the simulator.

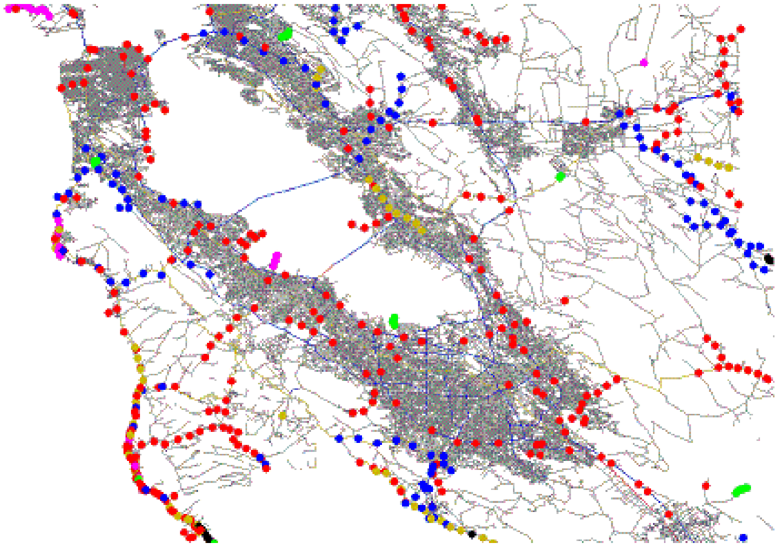


Figure 23: San Francisco simulator map

We chose the city of San Francisco, SF, for generating our data. The area of SF is about 47.5 square miles, and the population is 818,163 in 2010. There are about 470,481 registered vehicles, and 534,829 driving licenses issued by 2010. The number of jobs is about 437,073 jobs in 2010 and about 38.9% of employees prefer to drive alone to their work locations [75]. According to this data, we generated four datasets for the four location update mechanisms ALUM, EALUM, periodic, and silent period with random period of $[0:5]$ timestamps.

## 7.2 Choosing the Speed Threshold $SD_{Thresh}$

In our location update and pseudonym change mechanism, we suggest the update and pseudonym change under some conditions described in Section 6.4. According to these conditions, a pseudonym change and location update is triggered when the speed change of a vehicle exceeds the threshold value. The threshold value is to be determined according to the speed variations. The speed variance is calculated each timestamp and the value of the variance determines whether the movement is regular or irregular according to the value of $SD_{Thresh}$. The value of $SD_{Thresh}$ should be chosen carefully to best classify regular and irregular traffic conditions. Incorrect $SD_{Thresh}$ will result in an incorrect classification and may lead to lower performance in terms of quality and anonymity. For the best choice of $SD_{Thresh}$, we have first plotted the number of updates generated for different $SD_{Thresh}$ values as shown in Figure 23.

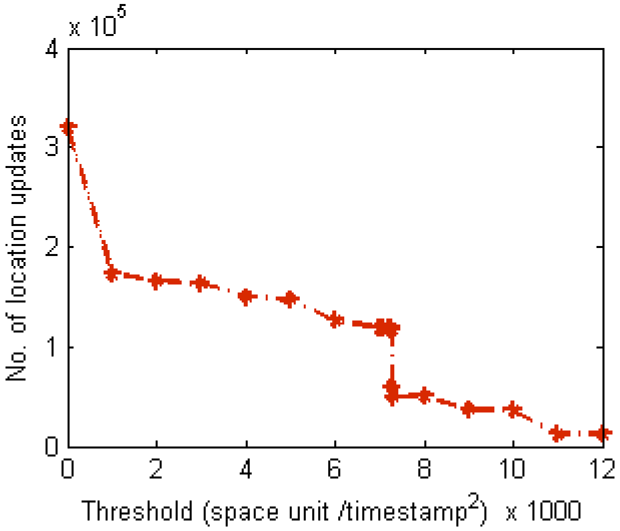


Figure 24: No. of location updates vs. $SD_{Thresh}$

The number of generated location updates from all vehicles can reflect regular or irregular traffic conditions. Under regular traffic flow, the curve in Figure 23 moves smoothly for low variance values and then performs a dramatic decrease before moving smoothly again. The point of this dramatic change is the best value of $SD_{Thresh}$. It separates regular and irregular traffic conditions. This point occurs between the values of 7000 and 8000 space unit/timestamp$^2$.

Determining the exact threshold classification value is done using the derivative of the curve of Figure 23. The data is fitted using Matlab and the fitted curve is then differentiated. The point where exact dramatic speed variance occurs is 7280 (space unit/timestamp$^2$) as shown in Figure 24.

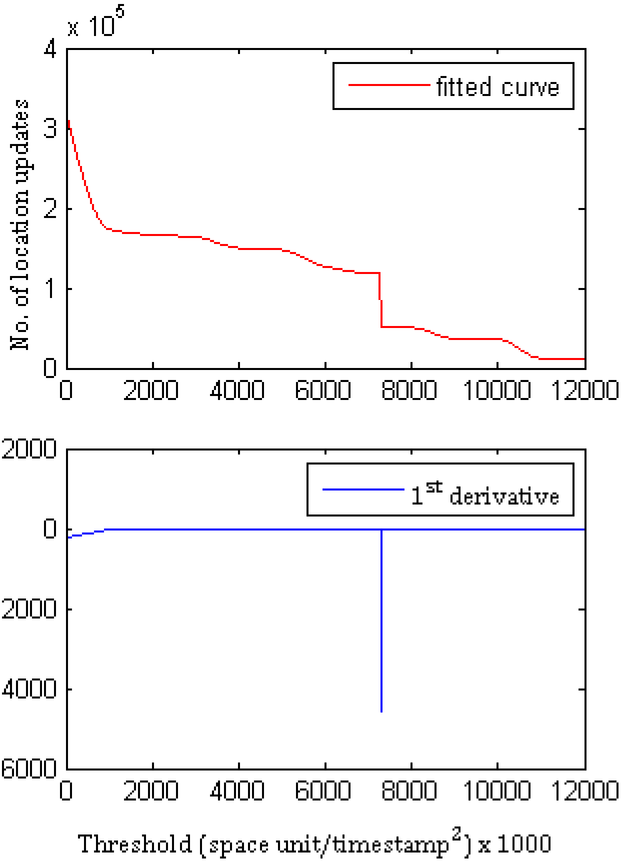


Figure 25: Determining the best $SD_{Thresh}$ value

Since different regions have different traffic activities, a global static $SD_{Thresh}$ may not be that efficient. Hence we suggest the use of a local static value by calculating

$SD_{Thresh}$ for each sub region. Users then use different $SD_{Thresh}$ values for the different sub regions they visit. In our work, we used a global $SD_{Thresh}$ value.

### 7.3 Choosing Sub Region Weight Threshold $W_{Thresh}$

Sub region weigh is used by EALUM to decide whether to update during silent period or not. By EALUM silent period, we mean the case where neither speed nor direction change occurs. When ALUM is to update according to the silent period condition, it first checks the weight of her sub region. If the weight is greater than a predefined weight, $W_{Thresh}$, then update is performed. Otherwise, update is delayed for another silent period. The selection of the value of $W_{Thresh}$ affects the level of privacy as well as the data quality.

We have divided San Francisco region into sub regions of 1km$^2$ each. Then we have calculated the weights of each sub region (as described in Section 7.3) for a time interval of 600 time stamps. The relative frequencies of the weights are shown in Figure 25. The values of weights range from 0 to 0.004.

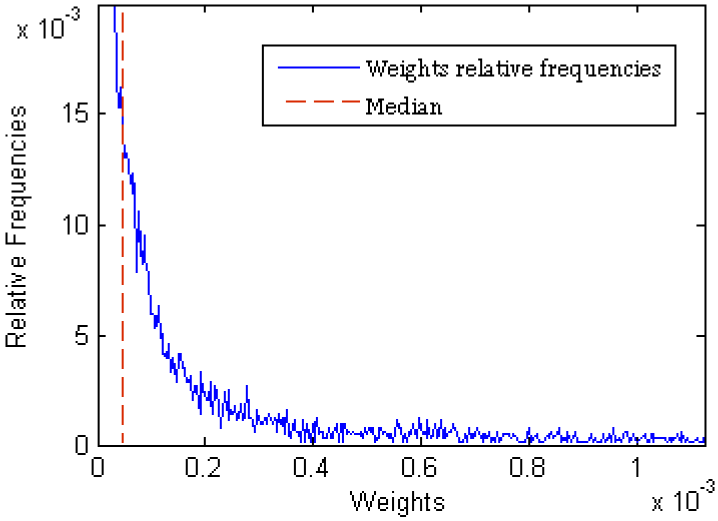


Figure 26: Relative frequencies of weights

We used the median value of the weights as $W_{Thresh}$ for the performance evaluation of ELAUM against the other location update mechanisms. However, we emphasize that $W_{Thresh}$ is to be set by the user herself according to her privacy preferences and the value here is for performance evaluations only.

### 7.4 Calculating $LocPred_{ERR}$

$LocPred_{ERR}$ is defined as the distance between the actual location of an object $A$ and its expected location. It is calculated as $|A.value(t + \triangle t) - (A.value(t) + A.function(t) \times \triangle t)|$, where $||$ denotes the absolute value function. We have calculated $LocPred_{ERR}$ for, ALUM, EALUM, periodic, and silent period. The percentiles of the distribution of $LocPred_{ERR}$ values are shown in Figure 26.

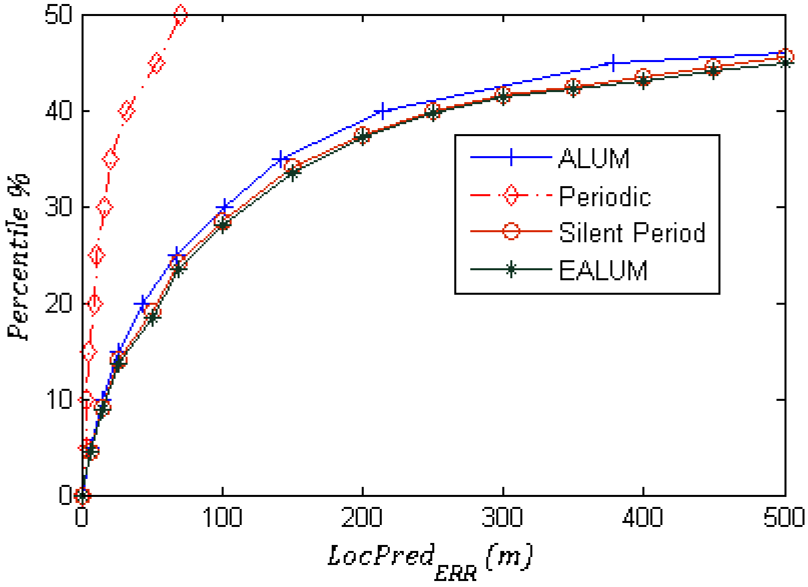


Figure 27: $LocPred_{ERR}$ percentiles

As shown in Figure 26, the frequencies of high $LocPred_{ERR}$ values in ALUM, EALUM, and silent period are higher than periodic. This means that their uncertainties regions will be larger and hence privacy will be better. An important factor affecting

$LocPred_{ERR}$ is the time gap between two successive updates. As shown in Table 6, the higher the time gap the higher the mean of $LocPred_{ERR}$.

Table 6: Mean location prediction error for moderate traffic

| | **Moderate daily traffic scenario** |
|---|---|
| **$\Delta$ t** | **Mean $LocPred_{ERR}$ (m)** |
| 1 | 18.90 |
| 2 | 37.51 |
| 3 | 55.76 |
| 4 | 73.62 |
| 5 | 91.07 |
| 6 | 107.90 |

For calculating the radius of uncertainties region, we will assume a strong adversary that can predict correctly the time gap, $\Delta$ t, between two successive updates, hence $LocPred_{ERR}$ will be chosen according to Table 6. For $GPS_{ERR}$, we will use the value of 14 meters [73].

## 7.5 $\boldsymbol{k-anonymity}$ Results

We have calculated the anonymity level for every point on each trajectory. The anonymity level was calculated as the number of location updates inside the uncertainty region, $UR$. The average $k$-anonymity level for the datasets is summarized in Table 7. From Table 7, we can infer that EALUM data is the best in terms of anonymity level $k$.

Table 7: Overall average anonymity level

| | **$Avg-k$** |
|---|---|
| ALUM | 5.575 |
| Periodic | 3.043 |
| Silent Period | 2.774 |
| EALUM | 6.19 |

The relative frequency of the occurrences of different *k* values is shown in Figure 27. From Figure 27 we see that both periodic and silent period behave almost similar while ALUM and ELAUM behave better for larger *k* values. It is also clear that EALUM is better than ALUM for even small *k* values. This puts EALUM on top of location update mechanisms in terms of privacy level.

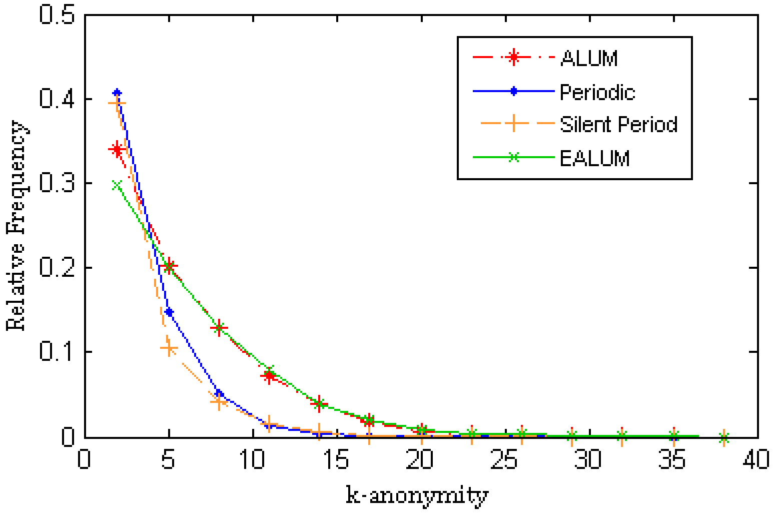


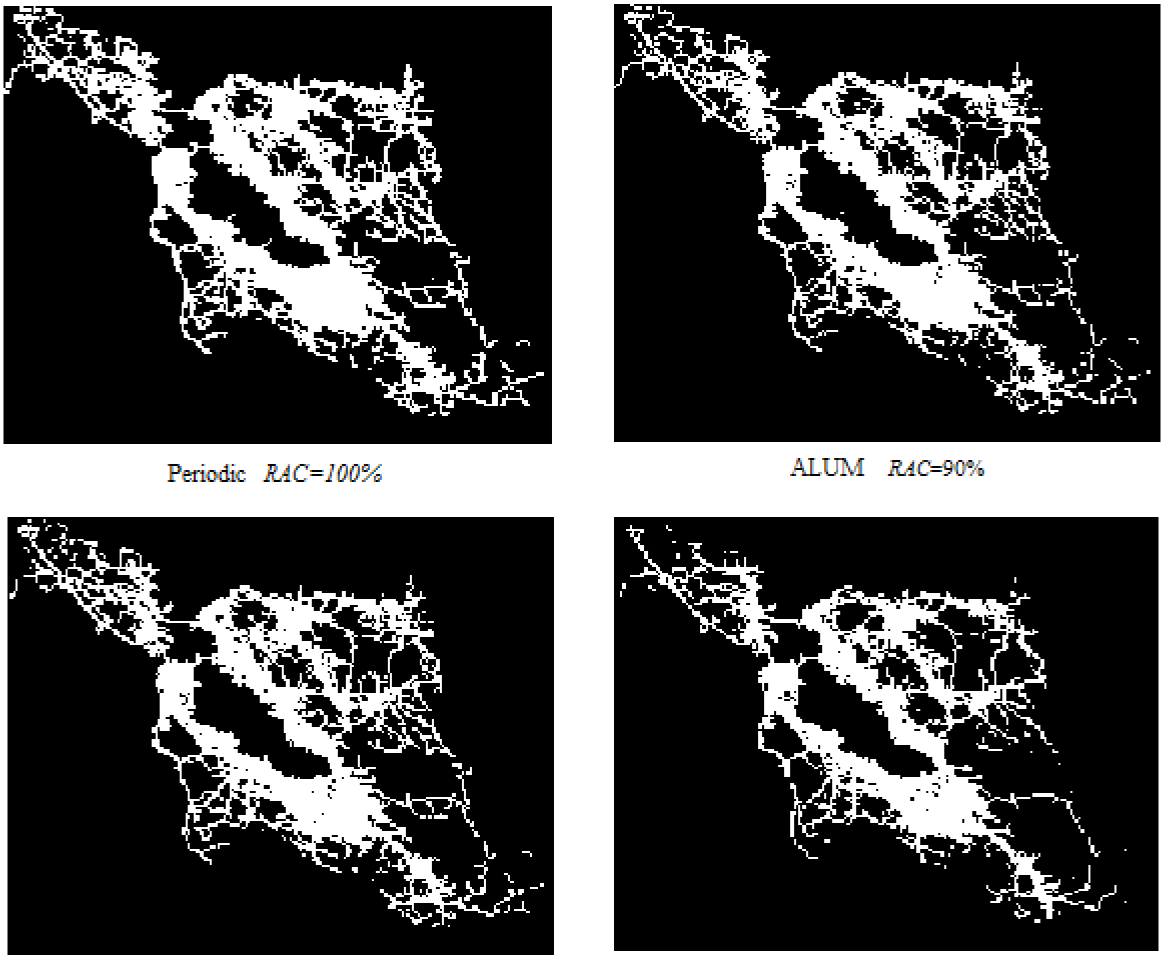
Figure 28: *k*-anonymity relative frequencies

## 7.6 Utility

Utility is defined as how far ALUM is useful for traffic monitoring. We used Relative Area Coverage (*RAC*), Weighted Road Coverage (*WRC*), and Relative Communication Cost (*RCC*) as utility metrics. The results of these metrics are reported in this section.

### 7.6.1 Relative Area Coverage (*RAC*):

We divided the area of San Francesco into blocks of 1 km$^2$ each. The number of location updates generated in a period of 60 time stamps is calculated for each block. Each block is then classified as covered if there are location updates in that region. Otherwise, the block is classified as not-covered. We have generated a black and white image (binary image) where each pixel represents a particular block. White pixels represent covered blocks where black pixels represent not-covered blocks. Relative Area Coverage, *RAC,* is calculated as the ratio of the coverage of each location update mechanism relative to the coverage of periodic location update mechanism. This is due to the fact that the best achievable coverage can be generated using periodic updates. Figure 28 shows the area coverage map for Periodic, ALUM, silent period, and EALUM location update mechanisms. *RAC* is reported below each image.



Periodic *RAC=100%*

ALUM *RAC*=90%

Silent Period *RAC*=87%

EALUM *RAC*=76%

Figure 29: Block coverage binary images

As shown from Figure 28, ALUM has an *RAC* of 90%. This means that ALUM can be used efficiently in traffic monitoring. EALUM is the worst in coverage, this is due to the fact of having less updates in lower weight sub regions. An important issue is that the many non-covered blocks face few traffic loads.

### 7.6.2 Weighted Road Coverage (*WRC*)

The weighted road coverage (*WRC*) metric was proposed by Hoh et al [24] to measure the utility of their uncertainty path cloaking mechanism. Traffic is monitored for a relatively long period of time over that region. The weight of each sub region is then calculated as the number of updates generated in that sub region divided by the total number of location updates performed in the total region using periodic update mechanism. Table 8 summarizes the results of *WRC* for the four location update mechanisms.

Table 8: *WRC* results

| **Update mechanism** | ***WRC* %** |
|---|---|
| Periodic | 100% |
| ALUM | 91% |
| Silent Period | 88% |
| EALUM | 83% |

EALUM is at the bottom of the coverage ratio with a pretty good value of 83%. The decision between ALUM and EALUM is a tradeoff between privacy and coverage. It is important to note that we used the median of the weights as the $W_{Thresh}$ value for EALUM. However, for privacy reasons users may adjust it to larger values and hence reduce the coverage ratio of EALUM.

### 7.6.3 Data Quality and Communication Cost

The number of location updates affects two utility parameters, data quality and communication cost. Generally, the lower the number of location updates the lower the quality of the data. On the other hand, the lower the number of location updates, the lower the communication cost required to perform these updates. The choice of these parameters is a tradeoff between the required quality of the data and the communication cost.

For traffic monitoring, the aim is to generate a current traffic map that covers well the area of interest. According to *RAC* and *WRC* metrics described in Section 7.6, ALUM and EALUM feed the traffic monitoring system with necessary data required for generating current traffic map. This means that ALUM and EALUM are suitable for traffic monitoring. However, in other applications that require higher data quality such as traffic prediction, ALUM and EALUM may not be sufficient and extra location updates may be required to fulfill the prediction requirements.

In traffic monitoring systems, the communication cost plays an important role in the successful of these systems. The penetration rate is defined as the percentage of vehicles carrying traffic monitoring equipment [24]. As the communication cost decreases, the number of subscribed users will increase. This in turn increases the penetration rate. An increase in the penetration rate results in an increase in the total number of location updates. This eventually increases the utility of the system in terms of coverage metric discussed in Section 7.6.

The communication cost is directly proportional to the number of generated location updates. The relative communication cost (*RCC*) is calculated as the total number of location updates in each location update mechanism divided by the total number of location updates using periodic update mechanism. Table 9 summarizes these results. As shown in Table 9, ALUM and EALUM decrease the communication cost significantly.

Table 9: *RCC* results

| **Update mechanism** | ***RCC* %** |
|---|---|
| Periodic | 100% |
| ALUM | 41.35% |
| Silent Period | 35.7% |
| EALUM | 34.4% |

It is important to remember that ALUM and EALUM have weighted relative road coverage of 91%, and 83% respectively. This significant decrease in communication cost while maintaining a very good coverage ratio makes ALUM and EALUM effective for traffic monitoring. The decision between ALUM and EALUM becomes an issue of privacy. If users prefer better level of privacy and more control on that level, then EALUM may be selected. On the other hand, for larger utility (area coverage) ALUM may be selected instead of EALUM since it has better coverage ratio.

## 7.7 Summary

In this chapter, we showed how to calculate both $LocPred_{ERR}$ and $W_{Thresh}$. We also calculated the anonymity level for the four location update mechanisms ALUM, EALUM, silent period, and periodic. The results showed that ALUM, and EALUM are better in terms of our privacy metric than periodic and silent period location update mechanism. Utility metrics showed that ALUM is sufficient for traffic monitoring with a significant decrease in communication cost.

# 8. CONCLUSIONS AND FUTURE WORK

Utilizing existing networks for the development of Collaborative Traffic Monitoring (CTM) is cheaper and needs no extra deployments. Therefore, it is more preferable than having a dedicated infrastructure network. However, such systems require more work for strengthening security and privacy. In this thesis, we proposed a privacy aware collaborative traffic monitoring system, PA-CTM, which adopts existing privacy solutions proposed for dedicated infrastructure, and works on existing networks. Users use pseudonyms to authenticate to traffic server and update their location or query current traffic status. They can also change their pseudonyms to enhance their privacy against the traffic server.

As the subtopics of this thesis, we firstly proposed a novel privacy-preserving pseudonym providing system, called CoRPPS (Collusion Resistant Pseudonym Providing System). In CoRPPS, several trusted entities are employed and the task of user authentication is split among several authentication servers. Other tasks and the corresponding user data are also split among trusted entities such that the collusion among them does not effectively link the real identity of a user to a pseudonym. This approach and the use of reusable tokens as anonymous identifiers in our design yielded high level of privacy for the users. The challenge of this design is that the link between the real user identities and pseudonyms should have been established by the request of law enforcement. In other words, there should have been a backdoor in the system, which contradicts the privacy requirements. We addressed this challenging issue in CoRPPS by enforcing all trusted parties to collaborate in the process of identity revealing. Analytical and simulation results showed that CoRPPS is applicable for different types of services; it can be tuned for different number of users according to the required level of anonymity, and the desired maximum collision probability. Our performance results also showed that CoRPPS is highly resistant against collusion attacks.

Our PA-CTM uses a novel autonomous location update mechanism, ALUM, which is managed by moving objects according to traffic conditions and hence does not require the existence of a trusted third party for controlling the location update mechanism. To enhance privacy, EALUM (Enhanced version of ALUM) utilizes traffic weights at different regions and perform location update and pseudonym change according to that. Our proposals ALUM and EALUM create a mix zone in an autonomous way. Experimental results showed that ALUM and EALUM can be used efficiently for traffic monitoring. The choice of ALUM or EALUM is a tradeoff between privacy and coverage. They both enhance privacy and reduce communication cost. However, ALUM is better than EALUM in terms of area coverage and the latter has a slightly better privacy results.

ALUM and ELAUM are effective for traffic monitoring based on synthetic datasets. It might be better if we applied ALUM and EALUM for real datasets. A comparison between our results for synthetic datasets and results of real datasets is intended to be done as a future work. For traffic prediction ALUM and EALUM may not be so efficient. As a future work, we intend to study efficiency of ALUM for traffic prediction and strategic planning. It may also worth studying incorporating other parameters for this purpose. These parameters include the road capacities, average speed, and daily time periods. Incorporation of such parameters requires detailed information about the regions of study.

# 9. REFERENCES

[1] B. Amro, Y. Saygin, and A. Levi, "P2-CTM: privacy preserving collaborative traffic monitoring," in Proceedings of the 3rd ACM SIGSPATIAL International Workshop on Security and Privacy in GIS and LBS San Jose, California: ACM, 2010.

[2] F. Dotzer, "Privacy issues in vehicular ad hoc networks," In proceedings of the Workshop on Privacy Enhancing Technologies, (2005).

[3] B. Hoh, M. Gruteser, H. Xiong, and A. Alrabady, "Enhancing security and privacy in traffic-monitoring systems," IEEE Pervasive Computing, vol. 5, pp. 38-46, 2006.

[4] "Total number of vehicles on register." vol. 2011 Australia: Australian Bureau of Statistics, 2011.

[5] "Intelligent transport", The Parliamentary Office of Science and Technology, London November 2002.

[6] RACQ, "The effect of Fuel Consumption & Vehicle Emmisions," 1/5/2008 2008.

[7] CARE, "Road Accidents Statistics in Europe, European Road Safety Day," April 27,2007.

[8] B. Halvorson: CNN, URL: http://www.cnn.com/2009/LIVING/wayoflife/05/22/aa.pay.as.drive.insurance, accessed February 2012.

[9] A. J. Blumbers and P. Eckersley, "On Location Privacy and How to Avoid Losing it For Ever," Electronic Frontier Foundation EFF, 2009.

[10] A. Gidari, "Locatio Privacy in a Mobile World," video lecture, URL: http://www.youtube.com/watch?v=YFo2VcfWCBQ, accessed February 2012.

[11] S. Lahlou, "Identity, social status, privacy and face-keeping in digital society," Social Science Information, vol. 47, pp. 299-330, Sep 2008.

[12] S. Mollman: CNN, Published 2008,URL: http://edition.cnn.com/2008/TECH/04/30/db.cellphonetracking/index.html . Accessed in March 2012.

[13] G. T. Patrick, "The Locus Opus: Playing with Privacy in a World of Ambient Intelligence," SSRN eLibrary, 2007.

[14] C. Cvrcek, M. Kumpost, V. Matyas, and G. Danezis, "A study on the value of location privacy," in Proceedings of the 5th ACM workshop on Privacy in electronic society Alexandria, Virginia, USA, 2006.

[15] B. Amro, A. Levi, and Y. Saygin, "CoRPPS: Collusion Resistant Pseudonym Providing System," in The Third IEEE International Conference on Information Privacy, Security, Risk (PASSAT '11), Boston, USA, 2011, pp. 1056-1063.

[16] B. Amro, A. Levi, and Y. Saygin, "Flexible, Fair, and Collusion Resistant Pseudonym Providing System, submitted to Computer Communications (COM COM) 2012.

[17] B. Amro, Y. Saygin, and A. Levi, "PA-CTM: Privacy Aware Collaborative Traffic Monitoring System Using Autonomous Location Update Mechanism," in 4th ACM SIGSPATIAL International Workshop on Security and Privacy in GIS and LBS SPRINGL 11, Chicago, USA, 2011.

[18] B. Amro, Y. Saygin, and A. Levi, "Enhancing Privacy in Collaborative Traffic Monitoring systems Using Autonomous Location Update," submitted to IET Intelligent Transport Systems, 2011.

[19] S. Zeadally, R. Hunt, Y.-S. Chen, A. Irwin, and A. Hassan, "Vehicular ad hoc networks (VANETS): status, results, and challenges," in Telecommunication Systems, 2012.(to appear)

[20] B. Aslam and C. Zou, "Distributed certificate and application architecture for VANETs," Military Communications Conference, 2009. MILCOM 2009.

[21] J. H. Song, V. W. S. Wong, and V. C. M. Leung, "Wireless Location Privacy Protection in Vehicular Ad-Hoc Networks," Mobile Networks & Applications, vol. 15, pp. 160-171, Feb 2010.

[22] C. T. Li, M. S. Hwang, and Y. P. Chu, "Secure and efficient communication scheme with authenticated key establishment and privacy preserving for vehicular ad hoc networks," Computer Communications, vol. 31, pp. 2803-2814, Jul 2008.

[23] H. Xie , L. Kulik , and E. Tanin "Privacy-Aware Traffic Monitoring," IEEE Transactions on Intelligent Transportation Systems, vol. 11, pp. 61-70, 2010.

[24] B. Hoh, M. Gruteser, H. Xiong, and A. Alrabady, "Preserving Privacy in GPS Traces via Uncertainty-Aware Path Cloaking," in Proceedings of the 14th ACM Conference on Computer and Communications Security, CCS'07, 2007, pp. 161-171.

[25] B. Hoh, M. Gruteser, R. Herring, J. Ban, D. Work, J. C. Herrera, A. M. Bayen, M. Annavaram, and Q. Jacobson, "Virtual Trip Lines for Distributed Privacy-Preserving Traffic Monitoring," in Proceedings of the Sixth International Conference on Mobile Systems, Applications, and Services, Mobisys'08. 2008, pp. 15-28.

[26] P. Belanovic, D. Valerio, A. Paier, T. Zemen, F. Ricciato, and C. F. Mecklenbrauker, "On Wireless Links for Vehicle-to-Infrastructure Communications," IEEE Transactions on Vehicular Technology, vol. 59, pp. 269-282, Jan 2010.

[27] X. D. Lin, R. X. Lu, C. X. Zhang, H. J. Zhu, P. H. Ho, and X. M. Shen, "Security in vehicular ad hoc networks," IEEE Communications Magazine, vol. 46, pp. 88-95, Apr 2008.

[28] G. Calandriello, P. Papadimitratos, J. P. Hubaux, and A. Lioy, "Efficient and Robust Pseudonymous Authentication in VANET," in Proceedings of the Fourth ACM International Workshop on Vehicular Ad Hoc Networks Vanet'07, New York, NY, USA, 2007, pp. 19-27.

[29] L. Buttyan, T. Holczer, and I. Vajda, "On the effectiveness of changing pseudonyms to provide location privacy in VANETs," in Security and Privacy in Ad-hoc and Sensor Networks. LNCS, 4572, Springer- Berlin, 2007, pp. 129-141.

[30] E. Fonseca, A. Festag, R. Baldessari, and R. Aguiar, "Support of anonymity in VANETs - Putting pseudonymity into practice," in 2007 IEEE Wireless Communications & Networking Conference, WCNC Hong Kong: IEEE, 2007, pp. 3402-3407.

[31] A. R. Beresford and F. Stajano, "Mix zones: User privacy in location-aware services," in Second IEEE Annual Conference on Pervasive Computing and Communications Workshop, 2004, pp. 127-131.

[32] M. D. Dikaiakos, A. Florides, T. Nadeem, and L. Iftode, "Location-aware services over vehicular ad-hoc networks using car-to-car communication," IEEE Journal on Selected Areas in Communications, vol. 25, pp. 1590-1602, Oct 2007.

[33] H. B. Hu, J. L. Xu, and D. L. Lee, "PAM: An Efficient and Privacy-Aware Monitoring Framework for Continuously Moving Objects," IEEE Transactions on Knowledge and Data Engineering, vol. 22, pp. 404-419, Mar 2010.

[34] B. Hoh and M. Gruteser, "Protecting location privacy through path confusion," in First International Conference on Security and Privacy for Emerging Areas in Communications Networks, Proceedings, 2005, pp. 194-205.

[35] C. Y. Zhang and Y. Huang, "Cloaking locations for anonymous location based services: a hybrid approach," Geoinformatica, vol. 13, pp. 159-182, Jun 2009.

[36] B. Gedik and L. Liu, "Protecting location privacy with personalized k-anonymity: Architecture and algorithms," IEEE Transactions on Mobile Computing, vol. 7, pp. 1-18, Jan 2008.

[37] C. Y. Chow, Chi-Yin, M. Mokbel, and X. 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 Arlington, Virginia, USA: ACM, 2006.

[38] "Availability of real-time traffic." vol. 2011: Google maps.

[39] M. Gerlach and F. Guttler, "Privacy in VANETs using changing pseudonyms - Ideal and real," in IEEE 65th Vehicular Technology Conference, 2007, pp. 2521-2525.

[40] A. Wasef and X. Shen, "REP: Location Privacy for VANETs Using Random Encryption Periods," Mobile Networks & Applications, vol. 15, pp. 172-185, Feb 2010.

[41] P. C. Johnson, A. Kapadia, P. P. Tsang, and S. W. Smith, "Nymble: anonymous IP-address blocking," in Proceedings of the 7th international conference on Privacy enhancing technologies Ottawa, Canada: Springer, 2007.

[42] R. Henry, "Nymbler: Privacy-enhanced Protection from Abuses of Anonymity." MSc Thesis: University of Waterloo, Waterloo, Ontario, Canada, 2010.

[43] P. Lofgren and N. Hopper, "BNymble More anonymous blacklisting at almost no cost," in Financial Cryptography and Data Security '11 St. Lucia 2011.

[44] Z. Lin and N. Hopper, "Jack: scalable accumulator-based nymble system," in Proceedings of the 9th annual ACM workshop on Privacy in the electronic society Chicago, Illinois, USA, 2010.

[45] E. J. Schwartz, D. Brumley, and J. M. McCune, "A Contractual Anonymity System," in Network and Distributed System Security Symposium NDSS, San Diego, CA, USA, 2010.

[46] P. P. Tsang, M. H. Au, A. Kapadia, and S. W. Smith, "Blacklistable anonymous credentials: blocking misbehaving users without TTPs," in Proceedings of the 14th ACM conference on Computer and communications security, Virginia, USA, 2007.

[47] E. Brickell and J. Li, "Enhanced privacy id: a direct anonymous attestation scheme with enhanced revocation capabilities," in Proceedings of the 2007 ACM workshop on Privacy in electronic society, Virginia, USA, 2007.

[48] P. P. Tsang, M. H. Au, A. Kapadia, and S. W. Smith, "PEREA: Towards practical TTP-free revocation in anonymous authentication," in 15th ACM conference on Computer and communications security, CCS ’08: Virginia, USA, 2008.

[49] D. Chaum, "Security without identification: transaction systems to make big brother obsolete," Communications of the ACM, vol. 28, pp. 1030-1044, October 1985.

[50] D. Chaum and J. H. Evertse, "A secure and privacy-protecting protocol for transmitting personal information between organizations," in Advances in Cryptology — CRYPTO’ 86: Springer Berlin 1987, pp. 118 - 167.

[51] J. Freudiger, M. Raya, and M. Felegyhazi, "Mix-Zones for Location Privacy in Vehicular Networks," in ACM Workshop on Wireless Networking for Intelligent Transportation Systems (WiN-ITS), Vancouver: Canada, 2007.

[52] Rongxing Lu, Xiaodong Lin, Tom H. Luan, Xiaohui Liang, and Xuemin (Sherman) Shen, "Pseudonym Changing at Social Spots: An Effective Strategy for Location Privacy in VANETs," IEEE transactions on vehicular technology, vol. 61, pp. 86 - 96 2011.

[53] J. Freudiger, M. Hossein Manshaei, J.-P. Hubaux, and D. C. Parkes, "On non-cooperative location privacy: a game-theoretic analysis," in Proceedings of the 16th ACM conference on Computer and communications security Chicago, Illinois, USA: ACM, 2009.

[54] G. Ateniese, J. Camenisch, M. Joye, and G. Tsudik, "A Practical and Provably Secure Coalition-Resistant Group Signature Scheme," in Proceedings of the 20th

Annual International Cryptology Conference on Advances in Cryptology: Springer-Verlag, 2000, pp. 255-270.

[55] D. Boneh and H. Shacham, "Group signatures with verifier-local revocation," in Proceedings of the 11th ACM conference on Computer and communications security Washington DC, USA: ACM, 2004, pp. 168-177.

[56] G. Ateniese, D. Song, and G. Tsudik, "Quasi-efficient revocation of group signatures," in Proceedings of the 6th international conference on Financial cryptography Southampton, Bermuda: Springer-Verlag, 2003, pp. 183-197.

[57] A. Kiayias, Y. Tsiounis, and M. Yung, "Traceable Signatures," in Proceedings of Eurocrypt 2004. LNCS. 3027: Springer-Verlag, 2004, pp. 571-589.

[58] M. Raya, P. Papadimitratos, and J. P. Hubaux, "Securing vehicular communications," IEEE Wireless Communications, vol. 13, pp. 8-15, Oct 2006.

[59] P. Samarati, "Protecting respondents' identities in microdata release," IEEE Transactions on Knowledge and Data Engineering, vol. 13, pp. 1010-1027, Nov-Dec 2001.

[60] L. Harris and Associates, "IBM multinationa consumer privacy survey," IBM, New York 1999.

[61] L. Rainie, J. Horrigan, A. Lenhart, T. Spooner, and L. Carter, "Trust and privacy online: why Americans want to rewrite the rules," Pew Internet and American Life Project 2000.

[62] S. Clauss and M. Kohntopp, "Identity management and its support of multilateral security," Computer Networks-the International Journal of Computer and Telecommunications Networking, vol. 37, pp. 205-219, Oct 2001.

[63] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: the second-generation onion router," in Proceedings of the 13th conference on USENIX Security Symposium - San Diego, CA: USENIX Association, 2004, pp. 21-21.

[64] M. K. Reiter and A. D. Rubin, "Crowds: anonymity for Web transactions," ACM Transactions on Information and System Security (TISSEC), vol. 1, pp. 66-92, 1998.

[65] S. Kopsell, R. Wendolsky, and H. Federrath, "Revocable anonymity," in Proceedings of the 2006 international conference on Emerging Trends in Information and Communication Security, Springer-Verlag, 2006.

[66] H. Ryan and G. Ian, "Formalizing Anonymous Blacklisting Systems," in Proceedings of the 2011 IEEE Symposium on Security and Privacy: IEEE Computer Society, 2011.

[67] N. Sebe, Y. Liu, Y. Zhuang, T. Huang, C. Hu, P. Liu, and D. Li, "A New Type of Proxy Ring Signature Scheme with Revocable Anonymity and No Info Leaked," in Multimedia Content Analysis and Mining. LNCS 4577: Springer Berlin / Heidelberg, 2007, pp. 262-266.

[68] J. Nadler, "Traffic Congestion and Air Quality," Association of Governments, Los Angeles 2007.

[69] P. Sistla, O. Wolfson, S. Chamberlain, and S. Dao, "Modeling and Querying Moving Objects," in Proceedings of the Thirteenth International Conference on Data Engineering: IEEE Computer Society, 1997.

[70] A. R. Beresford and F. Stajano, "Location privacy in pervasive computing," IEEE Pervasive Computing, vol. 2, pp. 46-55, Jan-Mar 2003.

[71] B. AMRO, A. Levi, and Y. Saygin, "CoRPPS: Collusion Resistant Pseudonym Providing System," in Third IEEE International Conference on Privacy, Security, Risk and Trust PASSAT 11 Boston, USA, 2011.

[72] E. Kaplan, T. B. Pedersen, E. Savas, and Y. Saygin, "Discovering private trajectories using background information," Data & Knowledge Engineering, vol. 69, pp. 723-736, 2010.

[73] J. S. Lewis, J. L. Rachlow, E. O. Garton, and L. A. Vierling, "Effects of habitat on GPS collar performance: using data screening to reduce location error," Journal of Applied Ecology, vol. 44, pp. 663-671, Jun 2007.

[74] T. Brinkhoff, "A Framework for Generating Network-Based Moving Objects," Geoinformatica, vol. 6, pp. 153-180, 2002.

[75] "San Francesco Transportation Fact Sheet," Municipal Transportation Agency, San Francesco November 2010.

[76] A. Shamir, "How to share a secret", Communications of the ACM, pages 612–613, 1979