Privacy-Preserving Location-Aware Data Availability and Access Authorization in Public Safety Broadband Networks

by

Hamidreza Ghafghazi

Thesis submitted to the
Faculty of Graduate and Postdoctoral Studies
In partial fulfillment of the requirements
For the Doctorate in Philosophy degree in Electrical and Computer Engineering

School of Electrical Engineering and Computer Science
Faculty of Engineering
University of Ottawa

Abstract

The increased demand for interoperability among Emergency Responders (ERs) and timely accessibility to a large amount of reliable, accurate, context and location aware, and privacy-preserved data (e.g., environmental data, health records, building plan, etc.), mandates the emergence of dedicated Public Safety Broadband Networks (PSBNs). However, realizing PSBNs and addressing such requirements encounters substantial challenges. For example, several security and privacy vulnerabilities have been detected in the Long Term Evolution (LTE) which is the leading enabler of PSBNs. Nonetheless, the more significant challenge lies under the corresponding data requirements. This is because data is unstructured, its volume is enormous, and it includes inaccurate, irrelevant, and context-free data. Moreover, the data sources are heterogeneous and may not be reachable in an emergency. Furthermore, the data contains personally identifiable information for which privacy and access authorization should be respected. In this thesis, we investigate and address the aforementioned challenges. Here, we propose an efficient and secure algorithm to mitigate the main security and privacy vulnerability of LTE. In addition, to provide context and location aware data availability during an emergency, we propose a secure data storage structure and privacy-preserving search scheme. Furthermore, we propose a location-aware data access model to filter irrelevant data with regards to an incident and prevent unauthorized data access. To envision our access model, we propose a location-aware fine grained access authorization scheme. Our security analysis shows that our search scheme is secure against a chosen keyword attack and the proposed authorization scheme is formally proven secure against a selective chosen ciphertext attack. Concerning performance efficiencies, our search scheme requires minimal data search and retrieval delay and the proposed authorization scheme imposes constant communication and decryption computation overheads. Finally, we propose a context-aware framework, which fully complies with emergency response requirements, based on the concept of trust to filter-out inaccurate and irrelevant data. The integration of our contributions promises highly reliable, accurate, context and location aware, and privacy-preserved data availability and timely data accessibility.

Acknowledgements

I would like to express my deepest gratitude to my supervisor Professor Hussein T. Mouftah for his outstanding guidance, support and motivation since I joined this program and throughout the writing of this dissertation. His meticulous supervision has not only educated me in research work but it will also serve as an inspiration and a model to follow in the future. Without his guidance and support, this dissertation would not have been possible.

Many thanks go to Dr. Amr ElMougy and Dr. Ala Abu Alkheir, my colleagues at the School of Electrical Engineering and Computer Science, who spent a considerable amount of their time with me in brainstorming ideas and approaches.

Dedication

This Thesis is dedicated to my parents.

For their endless love, support, and encouragement.

Table of Contents

List of Figures

List of Tables

List of Acronyms

4G	Fourth Generation cellular network
5G	Fifth Generation cellular network
ABE	Attribute-based Encryption
AKA	Authentication and Key Agreement
BDH	Bilinear Diffie Hellman
BDHE	Bilinear Diffie Hellman Exponent
BE	Broadcast Encryption
BF	Bloom Filter
CBF	Counting bloom filter
CCA	Chosen Ciphertext Attack
CCS	Central Cloud Server
CPA	Chosen Plaintext Attack
CP-ABE	Ciphertext Policy-Attribute-Based Encryption
C-RNTI	Cell Radio Network Temporary Identifier
CS	Cloud Servers
DA	Direct Authorization
DO	Data Owner
DoS	Denial of Service
DMT	Decision Making Tools
DPVS	Dual Pairing Vector Space
DT	Delay Tolerant
ECC	Elliptic Curve Cryptography

CKA	Chosen Keyword Attack
ER	Emergency Responder
FCR	Full Collusion Resistant
GO	Governmental Organizations
HE	Homomorphic Encryption
HIA	Hidden Identity Algorithm
HMAC	Hashed Message Authentication Code
HSP	Health Service Providers
HSS	Home Subscriber Server
IA	Indirect Authorization
IMEI	International Mobile Equipment Identity
IMSI	International Mobile Subscriber Identity
IoT	Internet of Things
IS	Index Server
ITS	Intelligent Transportation Systems
KGA	Key Generation Authority
KP-ABE	Key-Policy ABE
LA-CP-ABE	Location-Aware Ciphertext Policy-Attribute-Based Encryption
LSSS	Linear Secret Sharing Scheme
LT	Location Tracking
LTE	Long Term Evolution
MANETs	Mobile Ad Hoc Networks
MDT	Moderate Delay Tolerant
MI	Meta Information
MC	Mobile Cloud
MCC	Mobile Country Code
MKS	Multi-Keyword Search
MME	Mobility Management Entity
MNC	Mobile Network Code

MSA	MSIN Substitution Array
MSIN	Mobile Subscription Identification Number
MSP	Monotone Span Program
NDT	Non-Delay Tolerant
NGO	Non-Governmental Organizations
OBF	Obfuscating bloom filter
OE	Obfuscating Element
OPE	Oblivious Polynomial Evaluation
PBC	Pairing-Based Cryptography
PDA	Personal Digital Assistants
PE	Predicate Encryption
PEKS	Public-key Encryption with Keyword Search
PHR	Physical Health Records
PI	Personal Information
PII	Personally Identifiable Information
PIPEDA	Personal Information Protection and Electronic Documents Act
PKI	Public-Key Infrastructure
PPDR	Public Protection and Disaster Relief
PS	Public Safety
PSAP	Public Safety Answering Point
PSBN	Public Safety Broadband Network
QR	Query Router
RBF	Removal bloom filter
RIC	Random Identity Confidentiality
SA	Situational Awareness
SBF	Storage bloom filter
SEM	Smart Emergency Management
SER	Smart Emergency Response
SHMAC	Substituted HMAC

SMNs	Social Media Networks
SPDs	Smart Portable Devices
SSE	Searchable Symmetrical Encryption
TA	Trusted Authority
UE	User Entity
WPKI	Wireless PKI
WSNs	Wireless Sensor Networks
ZKP	Zero-Knowledge Proof

List of Symbols

1_T	Identity element in the target group
$\|\|$	Concatenation
$\vee$	Bitwise OR
$\xleftarrow{R}$, $\in_R$	Randomly choosing an element
$\bigcap$	Intersection of sets
Λ	A default attribute
$\emptyset$	An empty set
$\bigcup$	Union of sets
α, α'	Random number
β	Maximum number of files inserted in one buffer
γ	A location in PS_a
δ	A random number
ϵ	An Adversary's advantage to break a hard problem compared to random chance
θ	A Pseudonym
μ	A 0 or 1 value (flip of a coin)
η	Random nonce
ν	The result of the search scheme
ρ, ρ'	Random numbers
$\rho(i, j)$	A Map function
φ, φ'	A set of random numbers

ξ	An asymmetric encryption function
ξ'	A symmetric encryption function
ϖ	A specific overlapping probability
σ_u	The Memory index point to user u data
ς	A set of master keys
ϱ	Maximum number of attributes for an ER
τ_e	The number of pairing computations
τ_g	The number of group arithmetic operations
τ_p	Computational complexity of a pairing function
τ_t	Time interval t
ψ	Probability of a perfect match overlap
$\mathcal{A}$	An algorithm to break a hard problem
$\mathcal{B}$	A Challenger algorithm that uses $\mathcal{A}$
AA_{L_a}	Associated location area
$\text{AND}_+ - *$	AND gate with positive, negative, and wildcards
AND_m*	AND gate with multivalues and wildcards
B	A set of buffers
B_1	Buffer one
$Counter_{SH}$	Shuffling counter
C_i	Ciphertext i
C^*	Challenge ciphertext
C_{len}	Counter length
C_{new}, C_{old}	New and old ciphertext respectively
CS_x	Cloud server x
d_{def}	Number of default attributes
D_ξ	Decryption function
D	Access structure
e_{dist}	edit distance

$e(\cdot, \cdot)$	A pairing function
$E_{P_k}(Data)$	Public-key Encryption of data with the public-key P_k
E_t	Emergency occurrence time
E'_t	Current time
f	One way hash function
f_p	False positive rate
$\mathcal{G}$	Group generator function
$\mathbb{G}$	Bilinear group
$\mathbb{G}_T$	Target group
g	A generator of a cyclic group
g^x	An element of a cyclic group
G_i	Cyclic Group i
h_i	The i^{th} location area
H_i	Hash function
H_{K_s,P_s}	Hashed value
HA_u	Health attribute keywords set for user u
$K_{shuffle}$	Shuffling secret key
kw_i	The i^{th} keyword
K_X	A Secret Key for $X = \{1, 2, .., i, j, s\}$
l	Number of keywords/attributes
l_{AS}	Number of attributes included in an access policy
l_{HMAC}	Size of the hashed value
L_{W_i}	Location vector
L_i	Pseudo-identity of a location area
L_E	Emergency location
$\mathbb{L}$	set of all possible location areas
m	Length of a bloom filter
m_s	Size of the secret key and initial pattern arrays

m_{max}	The maximum size of allowed attributes associated with ciphertext
M	A message
M_{ir}	Identity response message
MSK	Set of private parameters
n'	Number of users
n_{att}	Number of attributes in the universe
n'_{rev}	Number of revoked users in the system
n	Number of location areas
n_c	Nonce
n_p	Number of primes that construct a composite-order group
n_S	Number of attributes in the system
N	Product of two distinct primes
N_W	Number of keywords
$O(n)$	Order n complexity
p	prime number
p_r	Probability value
P_k	A Public-key
$PEKS(P_k, KW_1)$	PEKS function with the public key P_k and the keyword KW_1 as input
$\Pr(\cdot)$	Probability function
p_{overl}	Probability of an overlap
P_s	initial pattern
PK	Set of public parameters
PS_a	Pseudo-identity for location area a
q	Maximum number of keywords in a users' index
q_{dj}	Number of distinct keywords in a disjunctive search query
Q	Query
$Q^s_{\mathbb{A}}$	Session key
r	A random number
r_u, r'_u	Random number

R	Random number
R_{match}	Number of matching results
PI_{em}	Emergency related personal information
RIC_l	Random Identity confidentiality length in bits
s	A random number
S	The set of users or locations for a broadcast encryption
S'	Ciphertext location set
S''	Preferred location set
S_{sec}	A security parameter
$\widehat{S}$	The set of users corresponding to a challenge ciphertext
$\|S_{KP}\|$	Secret keys and initial patterns sets size
$SK_{\mathbb{A}}$	Secret key (set) for user $\mathbb{A}$
SK_i	Secret key i
$Subs(X, Y)$	Substitution subroutine
t	A random number
t_{CR}	The threshold that guarantees the collusion resistance for BE
t_{reg}	Number of registered users
t_{sh}	Threshold access policy
$t_{i,j}$	Random number
T_c	Time stamp
T_{kw_i}	Trapdoor for keyword kw_i
$T_{i,j}$	A group member of G_{p_1}
U	Universe of attributes
$U_{P_{ID}}$	Pseudo-identity of a member of the system
V	Initial vector
$v_{i,j}$	Attribute value
Vw_i	The value associated to word i

W_i	Word/attribute i
W	A set of words or attributes of an ER
x	A random number
y_i	An element of the location vector
Y_0, Y_1, Y_2	Elements of the target group
z_i	An element i of the trapdoor vector
z'_u	A random number

Chapter 1

Introduction

1.1 Public Safety Networks

A Public Safety Broadband Network (PSBN) is the network used by Emergency Responders (ERs) for their communication needs. ERs include police, firefighters, and emergency medical care personnel. Current PSBN technologies (such as Project 25 and Terrestrial Trunked Radio) utilize narrow-band communication that results in limited interoperability among ERs and only supports voice and simple data communications [1]. However, dreadful experiences such as 9/11, the 7.0 magnitude Haiti earthquake in January 2010, the Boston bombing in April 2013, and the recent Paris attacks in November 2015 have shown the necessity of PSBNs in which interoperability among ERs and voice, video, and data communications capabilities are significantly enhanced [2, 3, 4, 5, 6, 7]. For example, the aftermath reports of 9/11 have indicated that lack of the ability to establish direct communications between different types of ERs caused uncoordinated emergency operation and even loss of many lives [5, 8, 9]. In that incident, many firefighters died because they could not receive the evacuation order that police officers received from their department. Besides, lack of sufficient awareness of the environment, the building dynamics, limited access to sensory data, limited real-time knowledge about the number of building occupants and their locations, etc., significantly suppressed effectiveness of the emergency response.

To facilitate interoperability and provide availability and accessibility of/to data such as environmental data, location data, maps, health records, and criminal records, etc., the Public Protection and Disaster Relief (PPDR) organizations have chosen Long Term Evolution (LTE) to be the leading technology for PSBNs [10, 11]. This decision is mainly because of characteristics of LTE such as simple all-IP architecture, flexible air interface with low latency, improved performance and efficiency, high reliability, and high capacity. From the security and privacy point of view, LTE offers mutual authentication, techniques to ensure freshness of various keys utilized for cryptographic algorithms, confidentiality of user plane data and confidentiality and integrity of control plane data and signaling for most parts of its security architecture [1, 7, 12, 13].

Using LTE and by employing smart mobile devices and wireless Internet access, ERs are able to directly communicate with one another and access the wealth of data at any time and location from diverse resources including governmental and non-governmental databases. This can help ERs achieve high levels of Situational Awareness (SA) during an emergency. SA is a human condition that enables an ER to understand his/her surroundings and react to its dynamic changes [14]. Consequently, this leads to a more effective emergency response.

1.2 Motivation

The interest in PSBNs is growing since they can significantly improve emergency response. However, the technologies and algorithms that are employed in PSBNs and for emergency response in general should match the requirements of PSBNs. Since PSBNs are applied for mission critical communications, their requirements are quite strict concerning high reliability, service and data availability, security, and privacy [3, 4, 5, 10, 11].

Regarding the communication technology, several security and privacy vulnerabilities have been detected in the LTE architecture [6, 7, 15, 16, 17, 18]. For example, a subscriber's identity confidentiality is not provided in all dimensions of the LTE. In this case, during the first attach process, in response to an identity request message sent by a Mo-

bility Management Entity (MME), a subscriber sends back his/her International Mobile Subscriber Identity (IMSI) in plaintext [6, 7, 19]. An IMSI is the permanent identity of a subscriber which could be sniffed by an eavesdropper in the preceding message exchange. This illustrates a potential breach of privacy. Observe that in commercial networks, it would be expensive for an attacker to perform this attack, and the result would simply be the identity of one regular subscriber. However, in PSBNs, this regular subscriber happens to be an ER, so the threat is largely magnified. Most of the solutions to address this vulnerability are based on public-key encryption that suffers from large communication and computation overheads.

Other than the security and privacy vulnerabilities of LTE, a key aspect of PSBNs' application (i.e., effective emergency response) is data availability. More precisely, effective emergency response requires accurate, relevant, timely, and context and location-aware data. The more data that is available to ERs, the higher the level of SA that is achievable for them [2]. However, acquiring such data encounters substantial challenges. First, there are unstructured and heterogeneous data sources, which indicates that data may be in many forms like text, photo, etc. [2, 3]. Second, there can be very large data volumes, e.g., 3.2 million tweets were sent in 24 hours after hurricane Sandy hit the US [20]. The obtained data should also be processed and filtered to become relevant information to prevent ERs from getting overwhelmed [3]. Third, unavailability of data sources because a disaster, like an earthquake, may destroy the communication infrastructure and data centers [2, 3, 8, 9]. Fourth, invalid data may be shared in an emergency, and the corresponding sources may be untrustworthy [20]. Fifth, the privacy of Data Owners (DOs) whose data is collected and processed, and authorized access to such data, are essential [20]. Sixth, identification and data retrieval should be done with as low a delay as possible [3, 8, 9, 20].

To address some of the aforementioned data availability challenges in the emergency response domain, we recognized two main approaches in existing products and in the literature. In the first approach, data collection, processing, and dissemination are taking place during an emergency. Well-known companies like Google [21], Facebook [22], and Microsoft [23] have announced specific products that enabled identifying missing people in

the recent disastrous incidents. Social media has been used in risk and crisis communication [24]. Microblogs such as Twitter have been utilized by the general public and ERs to share and disseminate data during catastrophic events like hurricane Sandy. Such data includes the situation of the affected area, the dynamics and progress of the situation, safety announcements, an individual's well-being and location, and so forth. The studies show that social media can improve cooperation between volunteers, emergency management officials, etc. [25, 26]. The credibility of data shared on Twitter for fourteen high impact events was analyzed in [27]. The authors showed that only 17% of the tweets comprising SA relevant data was credible. A graph-based data management system was designed to access and collect data from various social media sources to be used for emergency response [20]. The use of Twitter to broadcast data during a high impact event was studied in [28, 29]. These works concluded that the unreliable retweets were the fundamental problem the users faced during the disaster. Most of the studies and solutions in this approach are based on crowd-sourcing data for which data accuracy, trustworthiness, and privacy are remaining concerns [30].

The other approach considers foreseeing future emergencies occurrences. In this approach, individuals and organizations such the general public, Governmental Organizations (GOs), Non-Governmental Organizations (NGOs), and communities are encouraged to outsource their data to a storage system of their choice before an emergency occurs. Then, in an emergency, the basic sources of data would be those storage systems. As an example, the website Smart911 enables people to upload data about themselves such as their addresses, health conditions, family information, etc. [31]. When an individual calls 9-1-1, his/her data becomes available to the emergency dispatcher. This work suggests limited functionality since data access is only authorized if the caller is the one whose data is required. The authors in [32] propose an information system and construct a community-based virtual database gathering heterogeneous information from various resources for emergency management. A system was proposed to detect emergencies and enforces temporary access control policies in [33]. Such policies are defined in advance to bypass regular data access rules in an emergency to increase the availability of information. In general, the main shortcomings of the studies in this approach include the

complete reliance/trust of a DO on a server to handle his/her personal information, limiting data access authorization, lack of privacy-preserving context- and location-aware data availability and access authorization, and the need for filtering of the large volume of data.

1.3 Objectives

In this thesis, our goals are to protect privacy and enhance data availability in the context of emergency response. More precisely, we would like to propose a framework in which not only ERs' communications are secured and privacy-protected, but also they are enabled to retrieve a sufficient amount of data relevant to their emergency operations. The latter should be done in a timely fashion, and the retrieved data should be accurate, context- and location-aware, and privacy-preserved. Besides, to further empower privacy of DOs, a data access model should be enforced by a fine-grained location-aware authorization paradigm.

To accomplish our goals, our objectives in this thesis are summarized as the following. First, to understand the scope of the goals and challenges mentioned above, our preliminary objective is to study in depth the security and privacy vulnerabilities of the LTE architecture and the state-of-the-art algorithms that address data availability and access authorization in various domains such as E-health, emergency response, etc. Second, to propose a novel and efficient algorithm to enhance security and privacy of ER's communications by protecting the confidentiality of an IMSI in all signaling messages of LTE, e.g., during the first attach procedure. Third, to propose an innovative secure data storage structure and privacy-preserving context- and location-aware search scheme. This objective enables ERs to retrieve the required data during an emergency operation. Fourth, to propose a novel location-aware access authorization scheme. This empowers preserving the privacy of DOs in an emergency. Fifth, to propose an innovative framework to filter out irrelevant data in the context of emergency management. Such an objective enhances data accuracy during an emergency. The preceding proposed algorithm and schemes should also be carefully analyzed concerning security, privacy, and performance metrics so as to measure and validate their effectiveness in the context of the PSBNs application.

1.4 Contributions

The main research contributions of this thesis are summarized as follows:

- Proposed an efficient algorithm to protect the confidentiality of an IMSI in the LTE architecture.
- Proposed a novel secure data storage structure and privacy-preserving search algorithm to provide context and location-aware data availability in an emergency.
- Provided thorough security analysis and performance evaluations through comparative theoretical investigations and simulation results.
- Proposed a new emergency data access model which enables our access authorization scheme to be used as a data filtering technique. The model eliminates irrelevant data based on location and time of an emergency. Our emergency access model can also be interpreted as a new threat model which prevents unauthorized access with respect to the location and time of an emergency.
- Proposed a novel location-aware authorization scheme by integrating Broadcast Encryption (BE) with Ciphertext Policy-Attribute-Based Encryption (CP-ABE) in a novel way. The proposed scheme (called Location-aware Ciphertext Policy-Attribute-Based Encryption (LA-CP-ABE)) addresses the newly defined emergency data access model and mitigates the key escrow problem. In addition, the communication overhead and decryption computation complexity are constant regardless of the number of attributes in the access policy.
- Proposed a novel and context-aware framework to filter out inaccurate and irrelevant data using the concept of trust. The framework illustrates when, where, and how trust dynamics should be constructed and shows how trust evaluation should be applied in an emergency.

- Provided extensive security analysis and performance comparisons with the state-of-the-art solutions in the domain of emergency response and others to demonstrate the efficiency and effectiveness of the aforementioned proposed schemes.

1.5 Thesis Outline

This thesis consists of seven chapters outlined as follows: Chapter 2 offers the state-of-the-art algorithms protecting the confidentiality of IMSI, providing data availability, and access authorization in various domains such as emergency response and E-health. An algorithm is proposed to mitigate the permanent identity threat of the LTE architecture in Chapter 3. A secure and privacy-preserving scheme is proposed to provide data availability for emergency response in Chapter 4. Chapter 5 presents our novel location-aware authorization scheme pertinent for emergency response. Chapter 6 offers our proposed framework for trust in smart emergency management to filter out irrelevant data. Finally, Chapter 7 concludes the thesis and presents future research directions.

1.6 List of Publications

Journal Papers

- Ghafghazi, H., El Mougy A., Mouftah, H. T., Adams C., *Location-Aware Authorization Scheme for Emergency Response*, in IEEE Access, vol. 4, no. , pp. 4590-4608, 2016.

- Ghafghazi, H., Abu Alkheir A., Mouftah, H. T., *Trust in Smart Emergency Management*, Submitted to IEEE Intelligent Systems Magazine, November 2016.

Book Chapter

- Ghafghazi, H., El Mougy, A., Mouftah, H. T., Adams, C. *Security and Privacy in LTE-based Public Safety Network*, Wiley-ISTE - Public Safety Networks Series, 2016.

Conferences

- Ghafghazi, H., El Mougy A., Mouftah, H. T., Adams C., *Secure data storage structure and privacy-preserving mobile search scheme for public safety networks*, IEEE Wireless Communications and Networking Conference, Doha, 2016, pp. 1-7.

- Ghafghazi, H., El Mougy A., Mouftah, H. T., Adams C., *Classification of Technological Privacy Techniques for LTE-based Public Safety Networks*, in ACM Q2SWinet, Montreal, Canada, 2014.

- Ghafghazi H., El Mougy A., Mouftah, H. T., *Enhancing the Privacy of LTE-based Public Safety Networks*, in 13th Annual IEEE Workshop on Wireless Local Networks, Edmonton, Canada, 2014.

Posters

- Hamidreza Ghafghazi, Amr El Mougy, Hussein T. Mouftah, *Privacy techniques for LTE- based Public Safety Networks*, Wisense Workshop 2014, University of Ottawa, Aug 28, 2014.

- Hamidreza Ghafghazi, Amr El Mougy, Hussein T. Mouftah, *Privacy Issues and Challenges in LTE- based Public Safety Networks*, Wisense Workshop 2013, University of Ottawa, Aug 29-30, 2013.

Chapter 2

Data Availability and Privacy Preservation for PSBNs – State of the Art

2.1 Introduction

This chapter sheds light on some performance backgrounds and applications of the fourth and fifth generation of wireless communications (i.e., 4G (or LTE) and 5G respectively). In addition, a taxonomy of the required data for PSBNs will be presented. The proposed taxonomy shows the types of data that are required in various emergency situations. Afterward, the major security and privacy challenges and requirements of PSBNs are highlighted. Here, such challenges and requirements are divided into two parts; first, the challenges with LTE; second, the challenges that data requirements of PSBNs bring about. Furthermore, the state-of-the-art studies and solutions that addressed the aforementioned challenges will be reviewed, compared, and assessed against PSBNs' requirements.

The remaining sections of this chapter are as follows. Section 2.2 reviews important features of LTE, and presents the vision for further evolution toward 5G. Section 2.3 presents a taxonomy of different types of data required for PSBNs and identifies privacy concerns. The security and privacy challenges with regard to LTE are introduced and the

Table 2.1: LTE performance requirements for PSBNs

System performance	**LTE**
Operational frequency	700MHz
Bandwidth	10-20MHz
Control-Plane delay (Idle to connect)	<100ms
Control-Plane delay (Dormant to active)	<5ms
User-Plane delay	<5ms
Mobility	≤500kmph
Packet transport	All-IP transport
Physical layer modulation	DL: OFDMA, UL:SC-FDMA
Core network	Packet core with fixed /non-3GPP access

proposed solutions that address such challenges are reviewed in Section 2.4. Section 2.5 presents data availability challenges and requirements and classifies the respective state-of-the-art schemes with regards to security, privacy, and access authorization. Section 2.6 focuses on the state-of-the-art authorization schemes. The focus of this section is on the schemes that provide indirect authorization. Finally, Section 2.7 concludes the chapter.

2.2 LTE and beyond

PPDR organizations emphasize the need for dedicated PSBNs with the capability of providing a high level of reliability, service availability, interoperability, and security for critical communications [8, 3, 34]. Considering the preceding requirements, LTE has emerged as the leading candidate technology for PSBNs. Table 2.1 summarizes the LTE performance features. Among the features, user-plane and control-plane delays are significant since PSBNs require low latency signaling and data communications.

The demand for higher network capacity, spectral and energy efficiency, scalability, real-time service availability, communication reliability, etc., along with the exponential growth of mobile traffic have shaped the vision for the 5G wireless networks [35, 36, 37]. The main goals of 5G are to offer a hundred times higher user data rates and accommodate a hundred times more connected devices than LTE [38]. In addition, to consider real-time

services provisioning and to facilitate the next generation of Internet, end-to-end delay of 5G has to be less than 1ms while its spectral and energy efficiency need to be 10 times higher [39, 40].

It is expected that 5G will bring together all of the networks with different features and characteristics. In this regard, 5G will provide seamless and ubiquitous communications between any possible entities including humans and machines in any possible combination such as human to human, machine to machine, and human to machine. These communication paradigms are taking place at any time and any location [41]. This all-dimensional penetration of 5G into all of the entities and elements will create a user-centric information ecosystem [42].

Considering the preceding discussions, 5G is the future of the wireless communications to which PSBNs should be converging. The key reason for such a convergence is the fact that effective emergency response requires a large amount of reliable voice, video, and data communications. Using 5G, a wealth of data from heterogeneous sources such as building intelligence, Smart Grid, Intelligent Transportation Systems (ITS), Internet of Things (IoT), and Autonomous Vehicles is easily accessible [39, 43, 44, 45]. Unfortunately, PSBNs have been allocated between 10 to 20 MHz bandwidth in different countries such as Canada and US [10, 11]. With this bandwidth, the aforementioned requirement would not be met employing LTE [3, 46]. Despite the shortcomings in the performance of LTE, the current realization of PSBNs is going to benefit from this communication technology.

2.3 Taxonomy of Data required for PSBNs

One of the main motivations for the idea of PSBNs is the capability to access Data. Data availability is critical in achieving SA. Several types of data are required to achieve a proper level of SA. Among the vast amount of data, location data, sensory data, personal data, and available tools come to mind. This data could be merged, aggregated, or processed to obtain new information useful in various situations. However, private information is also among the aforementioned data for which privacy should be taken into consideration.

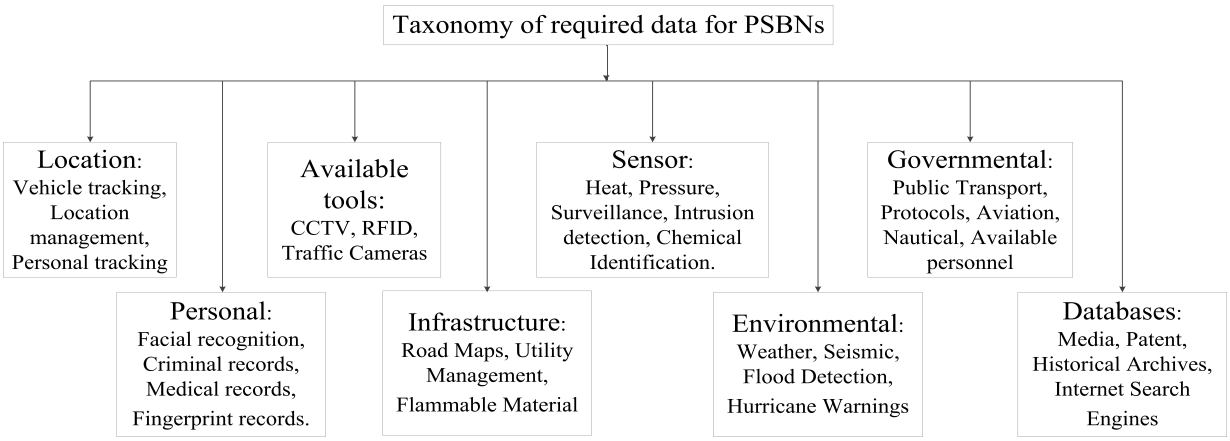

Figure 2.1: Taxonomy of required data for PSBNs

Before we proceed with different categories of data, the definition of personal information is given. Many countries, including Canada, have defined personal and private information and have set rules and regulations to determine how personal information should be treated. In the Privacy Act [47], for instance, personal information is defined as *information about an identifiable individual that is recorded in any form.* Similarly, in the Personal Information Protection and Electronic Documents Act (PIPEDA) [48] personal information is defined as *information about an identifiable individual.* This definition is a bit more general since the information does not need to be recorded to be considered personal information. Finally, in the technology environment, personal information is defined as *any piece of information which can potentially be used to uniquely identify, contact or locate a single person* [49].

Below, the required data for PSBNs are categorized. Figure 2.1 illustrates the taxonomy of such data.

Location Data: A Lot of the data that is required in various emergencies is about location. For example, the location of occupants in a building which is on fire is critical. Also, it is very critical that the location of the ERs is tracked to provide them with the right information at the right time about an incident. There is both public and private information in this category. For example, the identity of an ER that is tied to his/her location could be counted as private. In this case, only authorized entities should access such information. On the other hand, traffic data for a particular location could be counted

as public data.

Personal Data: This category involves data that uniquely identifies an individual such as medical records, criminal records, fingerprint records, facial recognition data, and so forth. For example, medical records could be of interest to firefighters so as to prioritize their rescue plans to help people with critical medical conditions first. In this category, the personal information should be accessed or processed only by authorized personnel.

Available Tools: The examples of available tools are Radio-frequency Identification (RFID) tags, Closed-Circuit Television (CCTV), traffic cameras, analytical and processing tools. For instance, the video stream of traffic cameras could be redirected to an ER to provide him/her with visual data of the current situation of the scene. Similar to the previous categories, personal information might also appear here.

Infrastructure Data: The infrastructure of the city, road maps, utility data, dangerous or flammable material and blueprints of the buildings are among the data in this category. For instance, medical emergency personnel could use information about the types of the toxic material that is present near the site of an incident. In this category, private information is minimal.

Sensor Data: Sensors generate different types of data such as temperature, pressure, and humidity. This data is very useful in overcoming the critical situations. For example, sensors used in a house to detect excessive heat or smoke generate valuable data. Moreover, utility data gathered by sensors in the Smart Grid communication network could be used by the firefighters to identify the source of the fire. For example, an excessive use of gas in a short period of time in a unit in a high-rise building can be an indication of a broken gas pipe. In this particular example, there exists identifiable information for which privacy should be preserved.

Environmental Data: Flood detection data, weather data, and hurricane warnings are among this type of data. This category could produce a great deal of helpful data for the ERs, for example, to plan their route to the incident. In this category, minimal or even no personal information may be found.

Governmental Data: Aviation data, public transportation data, construction and

public works, protocols and response guides and data about governmental personnel are included in this type of data. For instance, ERs should know about the protocols and guidelines that they need to follow in certain situations. In this category, there exists personal information such as the information about the governmental personnel.

Databases: This category contains several valuable sources of data. The Internet, media, and historical archives are among the popular databases. For example, historical archives could be used in particular regions to compare old data with contemporary data so as to generate more understanding of that location. Often databases, e.g., those accessible via the Internet, are full of personal information.

2.4 Security and Privacy Vulnerabilities of LTE

There are three main vulnerabilities in the LTE architecture that have been detected in the literature [6, 7, 15, 50]. In this section, we elaborate such challenges and review the state-of-the-art solutions that mitigate them. Then, we discuss and compare the solutions.

2.4.1 Subscriber Permanent Identity Threat

In the first attach procedure, the network asks for the permanent identity of a subscriber (i.e., IMSI) as depicted in Figure 2.2. In this figure, a User Entity (UE) and an MME are exchanging the identity request and response messages. A UE can be a mobile device or a communication device in general that is compatible with LTE. Since a security context has not been established between the network and the subscriber yet, the subscriber's IMSI is sent through the channel in plaintext [51]. IMSI is a permanent identity of a subscriber. Considering the preceding message exchange, a passive attacker, e.g. an eavesdropper, can easily sniff the IMSI. Such a disclosure may lead to information and location privacy breaches and Denial of Service (DoS) threats.

To solve the permanent identity threat, the authors in [17] offered to use dynamic mobile subscriber identity (DMSI) instead of an IMSI. After every authentication procedure, this

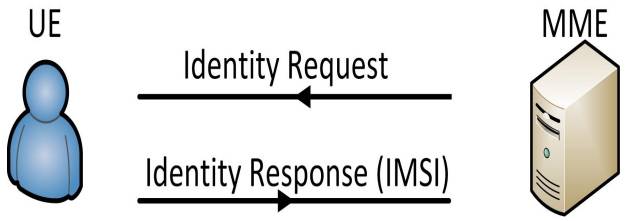

Figure 2.2: Identification procedure

value would be changed. DMSI is comprised of random numbers and specific permanent numbers as in (2.1).

$$DMSI = MCC||MNC||RIC||ERIC, \tag{2.1}$$

where the permanent numbers are Mobile Country Code (MCC) and Mobile Network Code (MNC). The rest are the numbers that are chosen uniformly at random. RIC stands for Random Identity Confidentiality and is used to uniquely identify a subscriber for a Home Subscriber Server (HSS). In addition, ERIC is the scrambled RIC. RIC has RIC_l-bit length, thus, the total number of available RICs is $2^{RIC_l} - 1$. Four RICs are allocated to each UE before deployment, which are Old, New, Pre, and Fresh RICs. UE is the subscriber's cell phone which contains an authorized SIM card. RICs are going to be changed every time the authentication procedure is required. A fresh RIC for every new authentication request is computed via a certain function. This is done to prevent the reallocation of the same RIC to one UE for two contiguous times.

This approach preserves the identity of the UE since the IMSI will never be transmitted through the channel. However, management of RICs needs additional processing effort and memory cost. Furthermore, the allocation of RIC to each UE from the network occupies excess bandwidth.

Public-key cryptography can be used to encrypt an IMSI [52, 53]. A UE uses the public-key of HSS to encrypt the IMSI. In this way, the privacy of the UE is preserved during the attach procedure. Public-key cryptography has also been used in [54], where a hybrid authentication, authorization, and key agreement protocol is proposed based on a

combination of a trusted model platform and Public-Key Infrastructure (PKI). The scheme uses passwords that are associated with a fingerprint and a public-key. Similarly, the work in [55] proposed an Authentication and Key Agreement (AKA) protocol using self-certified public-keys. The scheme authenticates a base station (which is called Evolved Node B (eNB) in LTE) based on the public-key broadcast protocol. Furthermore, the authors in [53] proposed an enhanced AKA in which they employ the Wireless PKI (WPKI) model and Elliptic Curve Cryptography (ECC). Finally, the works in [56, 57] use Elliptic Curve Diffie-Hellman to achieve AKA.

In order to facilitate identification of a subscriber, a password-based AKA is used in [58] to achieve a Zero-Knowledge Proof (ZKP) to protect the privacy of subscribers. In this case, MME still can perform identification, but the advantage is that this entity cannot learn anything about the identity. However, the computation cost of the scheme is high, especially for the handover procedure. The imposed delay of the algorithms that employ ZKP should be within the acceptable time interval to maintain the connectivity of the contiguous conversations.

2.4.2 Location Tracking

One of the issues that strongly violate privacy is Location Tracking (LT). LT simply indicates that a subscriber could be tracked while it is moving from one coverage area of the cellular network to another. Note that LTE assigns several different temporary identifiers to a single subscriber such as Temporary Mobile Subscriber Identity (TMSI), and Cell Radio Network Temporary Identifier (C-RNTI). In this regard, LT could be accomplished either by linking those identifiers to each other and to the permanent identities or by linking new and old temporary identifiers to one another. Thus, a successful LT attack relies on the ability to link subscriber's identities to each other.

An example is illustrated in Figure 2.3. Here, a passive attacker observing the radio channel may be able to link new and old C-RNTIs assigned to a subscriber within different cells if the sequence number by which the C-RNTI is computed, is consecutive. Note that C-RNTI is transmitted in plaintext. Therefore, it is crucial that the sequence numbers used

in the procedure of calculating C-RNTI is non-continuous. Authors of [7, 15, 51, 59, 60] suggested to protect the signaling of this procedure via cryptographic approaches. In addition, the use of periodic reallocation of C-RNTI in one cell was suggested in [15]. In this case, as the number of subscribers increases, the amount of signaling overhead rises which may lead to significant bandwidth consumption.

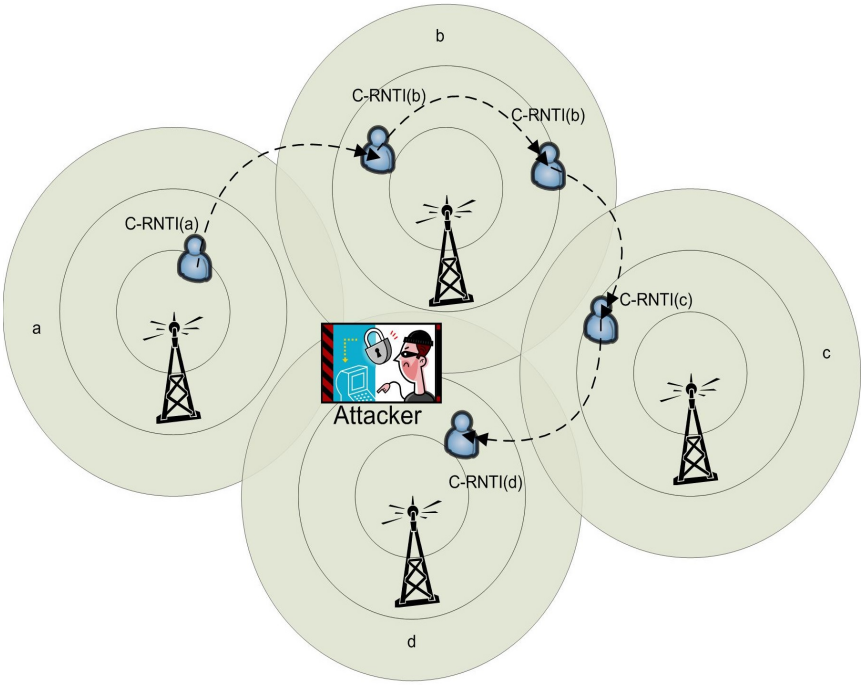

Figure 2.3: Location tracking attack using C-RNTI

The authors in [59] offered to apply the works that have been proposed for MANETs to preserve location privacy. One such example is to use mix zones which are areas where the temporary IDs of mobile equipment are interchangeable [61]. In this respect, the attacker would be misled and the location of devices would stay undetectable. However, a huge number of required signaling messages might still block fast and efficient intercommunication.

2.4.3 Paging Procedure in LTE

Another issue among security procedures of LTE arises when the network pages a UE. The paging process is as follows: A UE can be in different operational modes like active and

idle. When the UE is in the idle mode, it disconnects itself from the base station. Suppose the connection should be re-established with an idle subscriber as a result of a voice call initiation. The base station broadcasts a paging message within the subscriber's tracking area. This paging message contains a set of temporary IDs since the base station pages several subscribers at a time. The temporary ID that is included in the paging message is the TMSI which provides pseudonymity for the UEs [60]. Once the subscriber hears its TMSI, it will change its state to active and respond to the call.

Considering the paging procedure, suppose that an adversary is the one who has initiated the call by sending the request to the base station. Then, the attacker monitors the paging channel to obtain the set of TMSIs that have been paged by the base station within the subscriber's tracking area. Since there are several TMSIs within a single paging message, the attacker initiates the same call several times. Therefore, continuing this procedure would result in obtaining several sets of TMSIs for the attacker. At this point, intersecting those identities could yield the TMSI of the intended subscriber. The procedure is shown in Figure 2.4. Here, UE-2 initiates the paging attack by placing several contiguous calls to UE-1. The eNB broadcasts the corresponding paging requests in which UE-1's respective TMSI1 is included. After several rounds of paging requests, the attacker obtains TMSI1. Note that TMSI will not be changed within a particular tracking area and that the paging messages are not encrypted. Changing the tracking area by the subscriber would lead to obtaining a new TMSI. Thus, performing the same attack enables an adversary to track the location of the subscriber as well.

Note that in commercial networks, it would be expensive for an attacker to perform this attack, and the result would simply be the temporary identity of one regular subscriber. In the PSBNs, this regular subscriber is an ER. Therefore, the consequences of this particular attack may be much more serious.

To ensure privacy during the paging procedure, the authors in [60] proposed a physical layer approach. The authors used a UE's temporary ID as an input to a function. The output is a tag which is transmitted instead of a TMSI to page the UE. However, any correlation among the tags for different subscribers should not exist. The interesting point

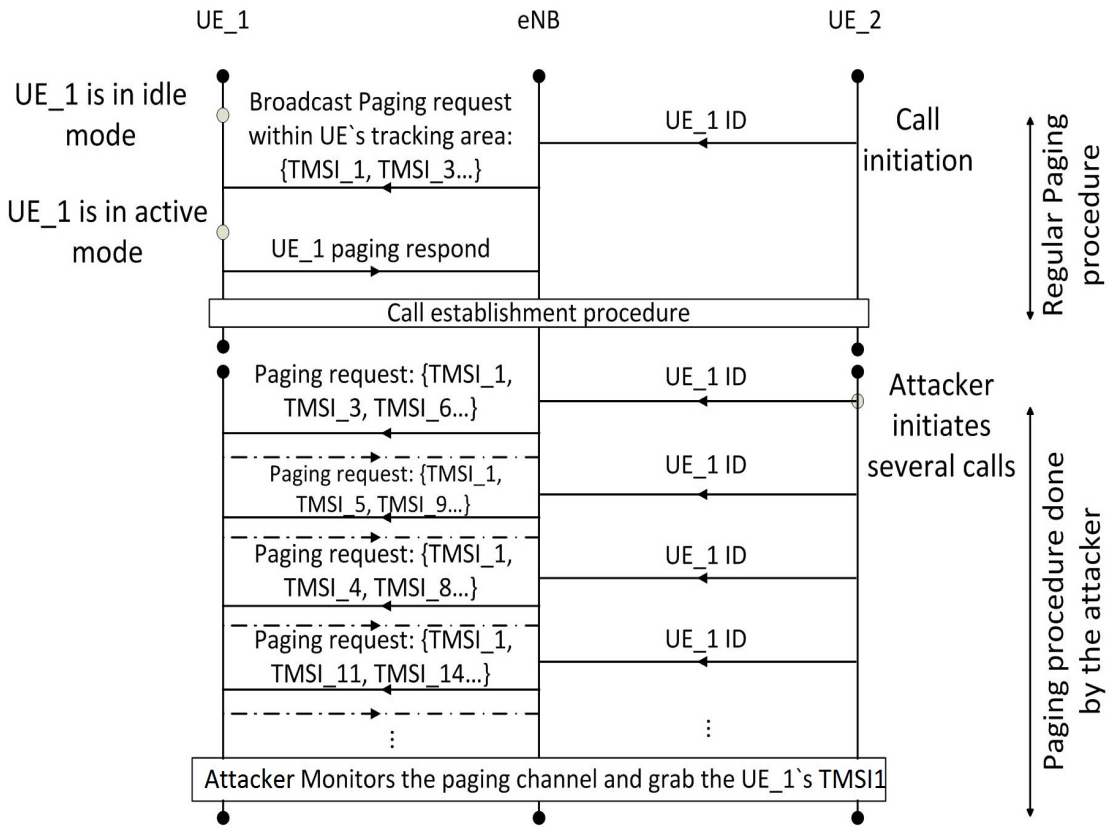

Figure 2.4: Paging procedure and the respective attack

is that the transmission power of the signal needs not to be at such a level that the receiver could decode it. The receiver should only be able to detect the signal to be able to ensure if she/he has been paged or not which results in saving energy. This scheme is also beneficial in terms of downlink bandwidth conservation. Despite the efficiencies of this approach, one drawback of it is the need to change the physical layer procedure that would lead to changing the hardware, which might be costly.

2.5 Data Availability for PSBNs

In this section, we classify state-of-the-art schemes that offer data availability. Here, our focus is on those schemes that inherently considered data confidentiality, and access authorization. Data confidentiality protects data privacy in data transmission, and upon data storage. In addition, access authorization prevents unauthorized data retrieval. Surveying

the literature, one can classify data availability into two approaches; centralized availability and decentralized availability. In the centralized availability, Data Owners (DOs) outsource their encrypted data to one/many cloud server(s) to which ERs should send data retrieval requests. However, retrieving proper data from a server containing encrypted data is challenging since the server does not understand which encrypted file contains relevant information that is requested in a query. Here, we review the schemes that provide search over encrypted data.

Considering decentralized data availability, on the other hand, in an emergency, DOs broadcast their encrypted data using smart mobile devices or Personal Digital Assistants (PDAs) to the users in their local proximity or to Health Service Providers (HSPs), Public Safety Answering Point (PSAP), and so forth to ask for help. The PDA monitors and collects health information using the sensors attached to the DO's body or the health information and other personal information are preloaded to the PDA or a smart card.

To achieve data privacy, in addition to data confidentiality, direct authorization or indirect authorization approaches are applied in both centralized and decentralized data availability. The direct authorization approach is used in private domains which are comprised of family, personal physician, friends, and neighbors, while the indirect authorization approach is applied in public domains that include researchers, healthcare personnel, other doctors, and ERs [62]. Considering the applicability of the indirect authorization approach in the emergency response, we also review state-of-the-art encryption schemes that inherently offer data access authorization in Section 2.6.

2.5.1 Centralized Data Availability

Considering centralized data availability, to protect confidentiality and privacy an encrypted version of data will be outsourced to a server. Note that the server cannot learn the content of data from the encrypted data. Then, to retrieve relevant data from the server, the schemes that enable search over encrypted data will be reviewed. In addition, both direct and indirect access authorizations are considered.

Direct Access Authorization

To achieve direct access authorization, any user who is interested in a DO's information should directly contact her/him even in an emergency and ask for access authorization. Here, access authorization is a preliminary stage of data retrieval and is included as a separate privacy protection building block to the encryption scheme. For example, in [63, 64], a DO sends decryption keys only if the user passes the authorization check phase. On the other hand, the most common way for direct authorization is to ask users to send search authorization requests to a DO before they send search queries to a server. In other words, to search over encrypted data, a user needs to generate an authorized search query for which he/she should contact the DO first. The following schemes are mostly following this method of direct authorization.

Song et al. [65] proposed a scheme that enables users to search over encrypted data. The result of the search for a word is the locations in the plaintext document where the word appears. The main idea of the scheme is, for every word in a document, the corresponding DO performs a pseudorandom function on the word to generate a value that is uploaded with the encrypted document to a server. A server can verify the existence of the word performing an XOR operation without knowing the queried word. This scheme has several features. First, anything that a server can learn about the plaintext is limited only to the result of the search process. More importantly, the queries are also hidden which means that the query word is kept secret from the database. However, this scheme cannot conceal the access pattern privacy. In other words, although queries are not known to the server, the database will still learn which portion of the data has been retrieved from the database. The computational complexity of the scheme for a document of size n is $O(n)$ stream cipher and block cipher operations.

Chang et al. [66] proposed an approach that enables DOs to privately search their outsourced encrypted data. Then, any user should get authorization from the DOs to be able to search the database. The key idea of this approach is to use keyword indexes that associate each keyword with the corresponding files in the database. The user conceals the keyword indexes within a pseudorandom bit string and sends the masked result to

the database. This bit string contains all keywords for which a user might search later. Then, to do the search process, the user sends a short query that enables the server to decode the corresponding portion of the index. In this case, the rest of the bit string remains pseudorandom. The access pattern is not protected in this approach. However, the data content is cryptographically protected. Concerning computational efficiency, the server should check all of the entries in the database for each search query (i.e., $O(n)$), test every file, and send back all of the files containing the intended keyword. This violates the database privacy since the user is able to retrieve more than one file in one session. This is regardless of the fact that the files are all in a ciphertext format.

Goh [67] proposed a method to produce secure indexes and use them to privately search over encrypted data. The secure index is a kind of data structure that only determines the presence of a word in the index using a clue. The clue is computed only using a secret key. The result of the scheme is a boolean value which is computed in $O(1)$ time per document. There are some features which make this work advantageous such as the ability to have variable keyword length and conjunctive queries. The security notion of this scheme is that the index does not leak any information about the content of the document. Moreover, the indexes of two different encrypted documents are formatted to represent the same number of keywords. This prevents a server to learn which document has more content than others.

Dong et al. [68] proposed a scheme that allows searching over encrypted data. The key idea behind this work is to use a proxy server to convert ciphertexts. For example, Alice encrypts data using her public key and outsources it to the proxy server. Then, Bob sends a request to recover that entry with the proof of authorization from Alice. Then, the proxy server converts the ciphertext in a way that Bob can be able to decrypt it using his own private key. This scheme also allows the users to search over the encrypted data using keywords. The server will convert a set of keywords to a new set which is encrypted over the general system parameters. In terms of security, the server has access to all of the messages and can decipher all of the transactions. In this regard, DO's privacy is invaded.

Raykova et al. [69] proposed a scheme that enables several parties to share sensitive data and can perform a secure and anonymous keyword search. This protocol provides

several security aspects such as query security, server privacy, server access control, and user anonymity. To achieve these features, the authors proposed to use two auxiliary servers called *Index Server (IS)* and *Query Router (QR)*. IS maintains the encrypted search structure using a Bloom filter. A Bloom filter is a data structure that utilizes a family of hash functions to insert a footprint of a word into an array of bits. Moreover, an IS protects the server's security and the user's anonymity. A QR is placed between users and the IS. It prevents other entities from learning anything about the identities of two communicating parties. The QR has another role which is a proxy server that converts the encrypted message of the user to a ciphertext suitable for the IS. To enable the preceding role and to provide efficiency, the scheme employs a deterministic encryption technique proposed in [70]. The assumption is that the encryption technique has the following group feature: $\xi_{K_1}(\xi_{K_2}(m)) = \xi_{K_1K_2}(m)$ where K_1, K_2 are the secret keys and ξ is the encryption function. The scheme imposes communication complexity of $O(R_{match})$ where R_{match} is the number of returned matching results.

Ballard et al. [71] proposed two schemes to perform conjunctive keyword searches on encrypted data. The components applied in these works are Shamir's secret sharing and pairing-based cryptography based on bilinear maps. In order to perform the private search, a DO first produces an index of each document. The index and an encrypted version of the document generated by a symmetric encryption scheme are stored in a server. Then, a user sends a clue to the server enabling it to search for a keyword. Each document is associated with a fixed size set of keywords.

Li et al. [72] proposed a fuzzy keyword search algorithm. The authors argued that the existing keyword search schemes require exact keywords set by a DO. However, in real situations, minor differences in choosing keywords by the querier might occur which result in negative responses by a server. To mitigate the preceding issue, the authors propose a fuzzy keyword search algorithm. The solution is based on the work on *Edit Distance* [73]. The edit distance between two words is defined to be the number of required instructions to convert one into another. Therefore, the server which has a set of keywords and predefined edit distance e_{dist}, upon receiving a query will check the queried keyword edit distance

with the set of keywords and if it was less than e_{dist} it will return the data; otherwise, it will not return anything.

Hore et, al. [74] proposed a method to securely perform multi-dimensional range queries over outsourced encrypted data. The key idea is to use secure indexes for the keywords. However, the authors suggested using bucketization techniques to divide the data into several partitions and compute appropriate tags from them.

The aforementioned schemes followed the centralized data availability with direct access authorization approach. Applying direct authorization is very confining in an emergency for two main reasons. First, it does not scale well. This means that there is a linear relationship between the number of data retrieval requests and the number of access authorization requests. In an emergency where the situation is critical and non-delay tolerant, this number can be enormous and the corresponding authorization delay may not be affordable. Second, DOs may be unconscious or may not even be reachable to grant access to the users in an emergency. Here, ERs achieve little situational awareness.

Indirect Access Authorization

Considering centralized data availability, indirect authorization has been used for the public domain in which DOs delegate access authorization to a cloud server. In this case, a user seeking particular data, without contacting DOs, sends a request to a server and retrieves the data all at once. Thus, this approach scales well which makes it more suitable for an emergency. Here, search query authorization is done by the server.

Boneh et al. [75] proposed a scheme called Public-key Encryption with Keyword Search (PEKS) that provides keyword search over encrypted data. In contrast with previous schemes, this work is based on public-key cryptography. With the aid of public-key cryptography, a DO can encrypt data with a user's public key and send the ciphertext along with some additional information to a server. Then, the user can search whether a particular keyword exists within the encrypted data or not. In this scheme, not only the data is encrypted but also the queries are all encrypted as well. Therefore, the only thing that is revealed to a server is the database access pattern. In this context, the only breach of the

user's privacy is the number of times that a certain encrypted record has been retrieved, or which record has not been queried yet.

The key idea of the scheme is that the user transmits a trapdoor T_{kw_i} which is related to a specific keyword KW_i to the server. Then, using the trapdoor the server finds all of the documents containing that particular keyword. The format of messages within the server is as (2.2),

$$\{E_{P_k}(Data), PEKS(P_k, KW_1), ..., PEKS(P_k, KW_l)\}, \tag{2.2}$$

where P_k is the user's public key and *PEKS* is the scheme that provides public-key encryption with keyword search. In this scheme, the authors used the Bilinear Diffie Hellman (BDH) intractability assumption which is as follows: given $\{g, g^x, g^y, g^z\} \in G_1$ and $e : G_1 \times G_1 \rightarrow G_2$ as the bilinear map, it is hard to compute $e(g, g)^{xyz}$. The computational complexity of this scheme is quite high. The pairing operations are all heavy operations compared to additive arithmetic. Therefore, if the number of keywords is large, the efficiency of the protocol drops drastically regarding computation delay.

Baek et al. [76] revisited the work in [75] and found out that it is possible for the server to gain information on the entire database using statistical analysis of the number of queries that it received for various parts of the database. Moreover, the authors eliminated the requirement of PEKS in which a secure channel between a user and a database should be established in order to transmit the trapdoors.

Abdalla et al. [77] also revisited [75] and showed that the protocol is statistically inconsistent. This is because the authors in [75] used a hashed value of keywords ($H(KW)$) in PEKS. However, since the keywords space is much larger than the hashed value space, thus, there is a possibility that $H_1(KW_1) = H_1(KW_2)$ where $KW_1 \neq KW_2$. This results in equal trapdoors for the two keywords. Hence, the outcome of PEKS shows false positive for such situations. In order to make PEKS statistically consistent, the authors proposed to use session keys and another hash function to increase the hashed value space.

Lie et al. [78] used PEKS to preserve keyword privacy. However, the scheme is not effi-

cient, firstly, because to retrieve proper information the entire database should be searched, and secondly, it is computationally expensive as PEKS employs Pairing-Based Cryptography (PBC). To tackle the latter, the authors in [68, 78] proposed to outsource the heavy computations of PBC to a proxy server. The approach converts a ciphertext in such a way that the decryption process is more lightweight at the user side. Despite the preceding improvement, in an emergency environment, the number of data outsourcing requests may be quite large because of a large amount of information requirement. This causes the delay to be increased. Furthermore, in such situations, the network infrastructure might be down which may result in the lack of access to the proxy servers.

Curtmola et al. [79] proposed Searchable Symmetrical Encryption (SSE). SSE is an encryption method that enables a DO to search over his/her encrypted data. However, this work enables other users to search over a DO's data as well which enables indirect authorization. To achieve that, the authors employed broadcast encryption along with traditional SSE. Broadcast encryption enables a user to encrypt a document for a group of legitimate users. Then, each member of the group can decrypt the document using his/her private key. The scheme has communication complexity proportional to the number of the documents that satisfy the query. The computational complexity for the user is $O(1)$. This scheme also does not provide access pattern privacy.

Tong et al. [80] propose that DOs delegate the access authorization to a private cloud. This scheme enhances [79] using pseudo-random number generators to avoid linkability of file identities. SSE uses linked lists in which file identities containing similar keywords are linked together in a secure way. The algorithm imposes minimum search delay since it does not need to search the entire database to find the result. However, its efficiency drops in dynamic situations in which files are added/removed to/from the system frequently. Also, the scheme is not able to perform multi-keyword search and the private cloud learns the keywords for which a user would like to search the database.

Kurosawa et al. [81] proposed an SSE which is secure against active adversaries. In most of the SSE algorithms, the server which holds the encrypted data and indexes is assumed to be trustworthy. In other words, the server will not perform arbitrary modification on the

stored data. However, a malicious server can modify data. This work proposed a verifiable SSE scheme in which the authors added a verification phase to [79]. This enables the user to verify the legitimacy of the data.

Cash et al. [82] proposed an SSE protocol that supports conjunctive search and boolean queries over encrypted outsourced data. This work performs search over arbitrarily-structured data and supports large database size. However, the scheme does not protect access pattern privacy. The main building blocks of this scheme are the works [79] and [83]. In addition, the authors use a function capable of performing oblivious shared computation between a user and a server such as the Diffie-Hellman based oblivious pseudorandom function. The scheme has also been extended to support dynamic changes such as an addition to the encrypted data [84].

Jarecki et al. [85] further extends the above algorithm to enable third party users who possess proper authorization from a DO to search over specific encrypted data. However, although the DO is the one who authorizes the third party users, the queries generated by the users are kept hidden from him/her. The main building block of the scheme is the work in [82]. The extension to enable indirect authorization is done using homomorphic signatures in which the DO signs the trapdoors and the third party user can later transform them into signatures of search queries.

Cao et al. [86] proposed a scheme which allows multi-keyword ranked searches over encrypted data. The key idea is to use a binary vector called an index vector for each document to represent the keywords within the document. Then, the search query which is also a binary vector can be compared with the index vector and similarities indicate the keywords in the document.

Bellare et al. [70] proposed a deterministic public-key encryption that enables fast searches over the encrypted data. Deterministic encryption has a particular limitation which is the fact that ciphertext contains partial information about the plaintext. This limitation can be mitigated if the partial information and the plaintext do not depend on the public key. The key idea behind this work is to associate a tag with a document and search for the tag in the data structure. Particularly, the scheme suggested that the tag is

a hashed value of the document.

Freedman et al. [87] proposed a method to privately access a database with the aid of keyword search. To construct the algorithm, the authors used Oblivious Polynomial Evaluation (OPE) with homomorphic encryption. The OPE used in this scheme is the *Paillier cryptosystem* [88]. This is because of its homomorphic property which is $\xi(KW_1) \times \xi(KW_2) = \xi(KW_1 + KW_2)$ where ξ is the encryption function. In addition, the paper proposed an approach to convert any keyword search algorithm which provides user privacy into a scheme which provides both user privacy and database privacy using oblivious evaluation of pseudorandom functions.

Chase et al. [83] proposed an algorithm to encrypt arbitrarily-structured data. The motivation behind this work is that most of the studies perform a keyword search over text-based data. However, not all of the data structures are text such as location information. The advantages of this work are twofold; first, an ability to perform private and expressive queries over encrypted data; second, providing authorized access to the data for other parties. The constructions proposed in this work are for a variety of data structures like labeled data, matrices, graphs, and so forth.

Gole et al. [89] proposed two schemes that privately search on encrypted data for conjunctive keywords. In other words, a user is able to perform a boolean combination of keyword search. The security of the scheme is based on a new model which says that the server should not learn anything beyond the result of the conjunctive query. Boneh et al. [90] proposed an algorithm that privately queries a database using somewhat homomorphic encryption. The queries are conjunctive. The key idea is to present a polynomial encoding of the database. Then, with the aid of somewhat homomorphic encryptions [91, 92], the scheme searches for conjunctive queries. To achieve that, the authors also proposed to divide the server into two servers called the server and the proxy. Similar insights were given in [69, 93].

Considering centralized data availability, to achieve indirect authorization, a DO can also enforce access authorization into the ciphertext using Functional Encryption schemes (for example, Attribute-based Encryption (ABE) or Predicate Encryption). This method

also eliminates the requirement of direct authorization where a user should directly contact a DO to obtain proper authorization. Barua et al. [94] proposed a scheme in which a DO sends data to an HSP along with the chosen access policy. To decrease the computation overhead on the DO, the authors proposed to delegate the process of incorporating the access policy into ciphertext to the HSP. The works [95, 96] use ABE and suggest to form an emergency version of encrypted data in which the owner only uses the "emergency" attribute to produce an emergency ciphertext. In an emergency, healthcare personnel can retrieve the emergency key to decrypt data. Li et al. [62] propose authorized multi-keyword search using Predicate Encryption. The delay corresponding to the search process is proportional to the size of the database, and it involves pairing computations. A detailed review of PSBNs pertinent ABE algorithms is given in Section 2.6.

2.5.2 Decentralized Data Availability

Considering decentralized data availability, in an emergency, DOs or their PDAs disseminate encrypted data either to the users within their local proximity or HSPs, PSAP, etc. Also, DOs may outsource their data to various servers and databases. For example, the data belonging to a DO might be stored in a hospital data server while the corresponding data might be stored in various public cloud servers such as Amazon and Microsoft for another DO.

In decentralized data availability, direct authorization is achieved by DOs checking users' legitimacy before data dissemination. If the users passed authorization checks, they would receive encrypted data and the decryption key. For example, the authors in [97] propose to opportunistically use authorized users to outsource health data processing in emergency situations. The authors propose a two-phase access control in which the first phase identifies medical users and the second phase uses a novel scalar product computation algorithm to ensure users' authorization. This approach involves three rounds of communications and pairing computation in the first phase of the check process.

To achieve indirect authorization, on the other hand, DOs enforce access policies into the ciphertext as mentioned above. For instance, in [98, 99] access control is encoded into

the ciphertext using ABE. The authors in [99] propose direct and indirect transmission modes. The latter delegates data transmission to a more powerful user. The authors in [100] consider different priorities for different types of health data and perform priority-based data aggregation and transmission. The scheme uses PBC for authorization checks and the Paillier cryptosystem for privacy-preserving data aggregation.

The common problem with the aforementioned schemes is the long imposed delay as the result of several rounds of communications [97] and heavy computational costs. In addition, the schemes are only applicable for individuals who have sensors attached to their bodies for health monitoring and need constant care. In an emergency, the endangered individuals might even be unreachable for some time intervals, and their PDAs or smartphones may be damaged. Therefore, these methods lack proper functionality.

2.5.3 Discussion

In this section, we classified the schemes providing data availability into two classes; centralized and decentralized. Since our focus is on the availability of data in an emergency for authorized users, we focused on the schemes that inherently respect data confidentiality. In other words, the data is transmitted to a user and stored in a server in ciphertext format. In addition, each class of data availability utilizes two different approaches for data access authorization, namely direct authorization and indirect authorization.

Table 2.2 summarizes the schemes that address data availability along with access authorization. The computational overheads of the schemes are elaborated in terms of the underlying foundation of the schemes, for example, Symmetric Cryptography (SC), PBC, and Hashed Message Authentication Code (HMAC). The schemes that use SC such as the AES algorithm and the ones applying HMAC require lower computation overhead than Asymmetric Cryptography (ASC) such RSA, Homomorphic Encryption (HE), and PBC. Besides, the schemes that offer search over encrypted data require either $O(1)$ or $O(n)$ computations imposed by the search process, where n is the database size. Considering the $O(n)$ complexity, the search algorithm checks every entry in the database to find the matches for the queried keyword while the $O(1)$ complexity directly returns back the entries

that contain the queried keyword.

Some of the search algorithms are able to perform Multi-Keyword Search (MKS) while the work in [72] provides Fuzzy Keyword Search (FKS). MKS can be used to reduce the communication overhead either from a user to a server or vice versa. The preceding advantage depends on the type of the query. For example, if the query is comprised of a set of conjunctive keywords, the server will only return the data that have the common keywords specified in the query. However, utilizing the single keyword search on the preceding query, the server returns all of the data that contain each keyword in the query. Then, the user finds the intersection of the received data. This example shows that conjunctive keyword search can decrease communication overhead from a server to a user. On the other hand, FKS allows queries to contain inaccurate keywords (i.e., the keywords that do not match completely with the ones associated with data). The shortcoming of the FKS is the false positive probability (i.e., it returns the data that may not contain the intended queried keyword). Finally, some schemes provide Anonymity (A) for the user using a proxy server. In this case, a server cannot learn which user is sending the data retrieval request. In Table 2.2, AP means access pattern privacy.

Applying centralized data availability in an emergency is challenging. This is because the centralized server may become a single point of failure. In addition, if the size of the database is large, utilizing the search schemes with $O(n)$ search computation complexity imposes a significant data retrieval delay. Moreover, none of the schemes considered location-aware data availability. Here, searching through the central database for a keyword (e.g., Asthma) without the inclusion of the context or location would result in data about both endangered individuals and the ones who may not even be in the emergency area. This is especially important in an emergency since such a shortfall prolongs the response time. Our proposed scheme in Chapter 4 is filling this huge gap.

In the centralized data availability, discussions regarding direct and indirect access authorization were given above. However, note that in most of the schemes under direct access authorization, the practice of authorization is applied in a system-wise point of view. In other words, the schemes can be transformed to indirect authorization schemes

Table 2.2: Data Availability Schemes Comparison

Scheme	Data Availability	Authorization Approach	Computation Overhead	Search Computation	Feature	Privacy Vulnerability
[65]	Centralized	Direct	SC	$O(n)$	-	AP, A
[62]	Centralized	Indirect	PBC	$O(n)$	MKS	AP, A
[63]	Centralized	Direct	PBC	×	-	AP, A
[64]	Centralized	Direct	PBC	×	-	AP, A
[66]	Centralized	Direct	SC	$O(n)$	-	AP, A
[67]	Centralized	Direct	HMAC	$O(1)$	MKS	AP, A
[68]	Centralized	Direct	HE	$O(n)$	Proxy Server	AP
[69]	Centralized	Direct	HE	$O(n)$	Proxy Server	AP
[70]	Centralized	Indirect	ASC	$O(n)$	-	AP, A
[71]	Centralized	Direct	PBC	$O(n)$	MKS	AP, A
[72]	Centralized	Direct	SC	$O(n)$	FKS	AP, A
[74]	Centralized	Direct	SC	$O(n)$	MKS	AP, A
[75]	Centralized	Indirect	PBC	$O(n)$	-	AP, A
[76]	Centralized	Indirect	PBC	$O(n)$	AP	A
[78]	Centralized	Indirect	PBC	$O(n)$	Proxy Server	AP
[79]	Centralized	Indirect	SC	$O(1)$	-	AP, A
[80]	Centralized	Indirect	SC, PBC	$O(1)$	Proxy Server	AP
[81]	Centralized	Indirect	SC	$O(1)$	-	AP, A
[82]	Centralized	Indirect	SC	$O(1)$	MKS	AP, A
[83]	Centralized	Indirect	SC	$O(1)$	MKS	AP, A
[85]	Centralized	Indirect	SC, HE	$O(1)$	MKS	AP, A
[86]	Centralized	Indirect	SC	$O(n)$	MKS	AP, A
[87]	Centralized	Indirect	SC, HE	$O(n)$	AP	A
[89]	Centralized	Indirect	ASC	$O(n)$	MKS	AP, A
[90]	Centralized	Indirect	ASC, HE	$O(n)$	MKS, AP	A
[94]	Centralized	Indirect	PBC	×	-	AP, A
[95]	Centralized	Indirect	PBC	×	-	AP, A
[96]	Centralized	Indirect	PBC	×	-	AP, A
[97]	Decentralized	Direct	PBC	×	-	AP, A
[98]	Decentralized	Indirect	PBC	×	-	AP, A
[99]	Decentralized	Indirect	PBC	×	-	AP, A
[100]	Decentralized	Indirect	PBC	×	-	AP, A

if the authorization is delegated to a server. In an emergency where DOs might be in danger, inaccessible, or even unconscious, a server will perform access authorization on their behalf. Although this transformation can be done in a straightforward fashion, it brings about another challenge which is the fact that the DOs should be trusting the servers. We will tackle this challenge as well in Chapters 4 and 5.

Applying decentralized data availability in an emergency is also cumbersome. As mentioned before, in decentralized data availability, the DOs themselves may disseminate their data during an emergency. As DOs might be inaccessible in an emergency or their smartphones and PDAs might be broken or lost, only relying on this approach is too risky. However, in a small incident where the number of endangered individuals is small and they are accessible, this approach is applicable. Here, the main limitation of such schemes is the fact that ERs should be near the individuals to be able to receive their data. In addition, if an individual's health condition is very critical so that data retrieval delay matters, some of the schemes are not applicable as they rely on heavy cryptographic algorithms or the authorization requires several rounds of communications.

In decentralized data availability models, different DOs may outsource their data to various databases. This increases the complexity of data retrieval during an emergency. Under such circumstances, limited identification information concerning DOs or servers is available. Without this information, one needs to send similar queries to all servers to retrieve the required information. This imposes a huge obstacle with respect to data availability. However, decentralized data availability is beneficial since it does not suffer from the single point of failure threat. Our proposed scheme in Chapter 4 addresses this huge gap. It also provides a location-aware search capability with constant computation time and very limited communication overhead.

2.6 Indirect Authorization Schemes

In this section, we review ABE and Broadcast Encryption (BE) schemes. ABE and BE schemes offer indirect authorization in which a DO encrypts data using a set of attributes

or identities respectively. To access data, considering ABE, a user should possess a subset of attributes satisfying the access policy. On the other hand, considering BE, a user's identity should be among the set of identities that the DO used to encrypt data. Using these schemes, a user does not need to contact the DO for access authorization at the time of data retrieval. This is especially appealing in an emergency situation.

Considering the emergency response domain, the performance efficiency of these schemes needs thorough investigation since they use heavy cryptographic arithmetic such as PBC. Considering an emergency, the schemes in this section utilize a Break-the-Glass concept in which a master key is provided to ERs to decrypt a ciphertext. Here, a DO forms an emergency version of encrypted data in which only an "emergency" attribute is used to produce the emergency ciphertext. This may not only enable unauthorized users to access information, but it may also overwhelm ERs by the large volume of accessible data. In Chapter 5, we propose a novel location-aware authorization scheme that authorizes users to access data belonging to individuals involved in an emergency if they are in the location area and at the time of the emergency.

2.6.1 Attribute-based Encryption

ABE is a relatively new authorization and public-key encryption technique which was first proposed by Sahai, *et al.* [101]. With ABE, an entity encrypts a message to some unknown receivers based on an access structure of his/her preference. However, the receivers are only able to decrypt the message provided that they possess a set of attributes satisfying the access policy. For example, Bob would like to share a document with certain individuals who are "Engineer and Manager". Note that the access policy for this example is an AND-gate. Alice has a set of attributes among which are engineer and manager. Therefore, she can decrypt the message from Bob. Note that any user who is able to satisfy this access policy can decrypt the message. Therefore, ABE is a valuable tool to provide authorization and confidentiality. There are two main types of ABE; CP-ABE [102] and Key-Policy ABE (KP-ABE) [103]. In CP-ABE, secret keys are associated with a set of attributes and ciphertext specifies the access policy. In KP-ABE, the ciphertext is associated with the set

of attributes and the access policy is encoded into the secret key of a user. In this work, we only focus on CP-ABE as it provides more control over who can have access to data in comparison with KP-ABE.

We categorize CP-ABE schemes based on various access policies. The first sub-category is comprised of schemes which offer flexible and expressive access policies. Here, the schemes rely on a monotone access tree structure supporting AND-gate, OR-gate, and threshold [104]. These schemes use a secret sharing scheme such as Shamir's secret sharing [102, 105, 106]. In addition, some schemes utilize Linear Secret Sharing Scheme (LSSS) facilitating the conversion of any boolean formula into an LSSS representation (i.e., Monotone Span Program (MSP)) [104, 107, 108, 109]. In both cases, the encrypting party chooses a secret and shares it among the attributes in the access policy following the secret sharing paradigm and generates ciphertext. The ciphertext size in these schemes grows linearly with the number of attributes in the access policy. In addition, the computation complexity of the decryption process in such schemes depends on the number of attributes satisfying the access policy. Therefore, it can be seen that there is a trade-off between the expressiveness of an access policy and efficiency of the scheme in terms of communication overhead, computation complexity, and delay. The more expressive an access policy is, the less efficient the CP-ABE scheme becomes.

On the other hand, the second sub-category is comprised of protocols with lower flexibility and expressiveness for the access policy. In theses schemes, the access policy does not support OR-gates in particular. Here, the schemes support AND-gates access structure and threshold access structure [110, 111, 112, 113]. The attributes may have a single positive value, both positive and negative values, or multiple values (e.g., +1, -1, (2, 3, -5, ...) respectively). In addition, some schemes provide wildcards in the access structure which mean that an attribute can have any value in its allowed range. There are schemes in this group where the ciphertext size depends on the number of attributes in an access policy [114]. However, the majority of the schemes have constant ciphertext size regardless of the number of attributes in the access policy. In addition, a large number of constructions offer constant decryption computations. In this case, it is particularly important to have

a constant number of pairing operations as this is the dominant factor of computation complexity and delay. Constant communication and computation costs are attractive to critical applications in which resources are constrained and low delay is of significance.

Emergency situations are highly dynamic in which the time and location of data access are varied. Therefore, such dynamic features make the use of CP-ABE complicated. To the best of our knowledge, a concrete CP-ABE scheme that incorporates dynamic attributes into the ciphertext has not yet emerged. We will elaborate corresponding challenges and requirements in detail in Section 5.4. On the other hand, CP-ABE schemes need a Trusted Authority (TA) to compute the secret keys. In this case, the problem of key escrow rises in which the TA is able to decrypt every encrypted document. Our proposed scheme mitigates such an issue. Table 2.3 shows different features of CP-ABE schemes. In Table 2.3, Del means delegation of secret keys, Rev means revocation of users or attributes, and PNW means positive negative wildcard.

.

Table 2.3: ABE protocol comparison

Scheme	ABE type	Security	Access Policy expressiveness	Constant Ciphertext Size	Constant Decryption Computation	Other features
[102]	CP-ABE	Full(CPA-GG)	Tree, Shamir	×	×	Del, Rev
[104]	CP-ABE	Selective	MSP(LSSS)	×	×	-
[105]	CP-ABE	Selective	Tree, Shamir	×	×	Del
[106]	CP-ABE	Full(CPA)	Shamir	✓	×, Constant pairings	-
[107]	ABE	Full(CCA2)	MSP(LSSS)	×	×	-
[109]	Broadcast ABE	Selective	MSP(LSSS)	×	×	Rev., Del.
[108]	CP-ABE	Selective	LSSS	×	×	Multi-authority
[110]	CP-ABE	Selective	Multi-value AND-gate	✓	✓	Short secret key
[111]	CP-ABE	Full (CPA)	Multi-value AND-gate	✓	✓	Hidden policy
[112]	CP-ABE	Selective	Threshold AND-gate	✓	×, Constant pairings	-
[113]	CP-ABE	Selective	AND-gate with PNW	✓	✓	-
[114]	CP-ABE	Selective	Multi-value AND-gate	×	×	-

2.6.2 Broadcast Encryption

BE enables a broadcaster to encrypt a message for some subset S of users in a system with a total of n' users. In this regard, any user in S uses his private key to decrypt the ciphertext. However, users outside of S cannot learn any information from the ciphertext and cannot collude with each other to decrypt the message. Such a feature makes a BE scheme collusion resistant. Applications of BE are several such as key distribution and secure distribution of copyright media [115, 116, 117, 118].

In BE, it is preferable that the following features are achieved: the system is public key which means that anyone can broadcast ciphertext; receivers are stateless which means that they do not need to update their private keys, and a BE is collusion resistant against all users outside the selected set S [119]. Note that BE schemes consist of two main parts. One part uses a group secret key as an input to a symmetric encryption scheme such as AES and encrypts the message with that. The other part is the actual BE scheme which is a public key scheme to broadcast the group secret key. Then, receivers decrypt the BE message (i.e., the group secret key) first, then use that as the input to the symmetric encryption scheme to decrypt the actual message.

Fiat *et al.* proposed the first formal BE with $O(t \log^2 t \log n')$ ciphertext-size where n' is the total number of users in the system and t_{CR} is the threshold that guarantees the collusion resistance of the scheme [120]. Naor *et al.* proposed a fully collusion resistant BE scheme [121]. The scheme broadcasts a message to all users except a small set n'_{rev} of revoked ones. The ciphertext size of the scheme is proportional to $O(n'_{rev})$, but the private keys are of size $O(\log^2 n')$. The works in [122, 123] decreased the private key size of the scheme to $O(\log n')$. Selvi *et al.* also proposed a fully collusion resistant BE scheme [124]. In such works [122, 121, 123, 124], the ciphertext size grows linearly with the size of the receivers set (i.e, $|S|$), or the number of revoked users $|n'_{rev}|$. However, Boneh *et al.* proposed two fully collusion resistant BE schemes [125]. The size of the ciphertext in the first construction is constant and for the second one is $O(\sqrt{n'})$. The scheme applies bilinear maps to achieve the ciphertext size for both schemes. However, the scheme is based on the selective security model in which the security proof is done with a prior step called

Table 2.4: BE protocol comparison

Scheme	Communication overhead	Computation complexity	Security
[120]	$O(t \log^2 t \log n)$		$(t, n)-$CR
[121]	$O(r')$	$O(\log n')$	FCR
[122]	$O(r')$	$O(\log n')$	FCR
[123]	$O(r')$	$O(n')$	FCR
[124]	$(S+4)\|G_1\|$	$2\tau_e + S\tau_g$	FCR, Adaptive
$[125]_{1,2}$	$\{2\|G\|\}_1, \{O(\sqrt{n'})\}_2$	$2\tau_e + S\tau_g$	FCR, Static
[126]	$\|G_1\| + \|G_2\|$	$2\tau_e + r\tau_g$	FCR, Static
[119]	$2G_1$	$2\tau_e + S\tau_g$	FCR, Adaptive
[127]	$O(\sqrt{n'})$	$4\tau_e + \|S+1\|\tau_g$	FCR, Adaptive

initialization. In such a step, an adversary chooses the target set, $\widehat{S} \subseteq S$, corresponding to his/her challenge ciphertext. Similarly, Gentry *et al.* [119] proposed a BE scheme which is secure against an adaptive attacker meaning that the attacker can send any set $\widehat{S}$ of challenge ciphertexts and the initialization step is eliminated from the proof. Delerablée *et al.* proposed a dynamic BE scheme in which there is a Join operation that alters public keys to address such dynamicity [126]. The ciphertext size and private key size of the scheme are constant. Boneh *et al.* proposed another fully collusion resistant BE scheme which has $O(\lambda\sqrt{n'})$ ciphertext-size where λ is the security parameter [127].

Considering an emergency, the communication overhead and computation complexity of BE schemes should satisfy the requirements of an emergency. Table 2.4 summarizes the aforementioned BE schemes. In this table, FCR means Full Collusion Resistance; τ_e and τ_g represent the number of pairing computation and group arithmetic operations respectively and are the dominating sources of computation delay. For communication overhead, we merely show the elements that are the points of difference in varied schemes. In other words, we neglected the ciphertext element representing the output of a symmetric encryption scheme.

2.7 Conclusion

In this chapter, we highlighted some security and privacy features of LTE and identified respective challenges. The state-of-the-art solutions to mitigate such challenges were reviewed and compared. We also shed light on the main challenges of PSBNs with regard to data requirements. A taxonomy of the required data for PSBNs was presented and privacy concerns were identified. State-of-the-art private search algorithms and indirect authorization schemes were reviewed and compared. In addition, ABE and BE schemes were surveyed as methods of indirect access authorization and their application in an emergency was assessed.

Chapter 3

Enhancing Privacy within the LTE Architecture

3.1 Introduction

In Chapter 2, three main security and privacy vulnerabilities of the LTE were identified. Among the three, the subscriber permanent identity (i.e., IMSI) vulnerability is more challenging. This is because the permanent identity of a subscriber is transmitted in plaintext format in certain signaling messages such as the first attach request. The main goal of the attach request is to facilitate the mutual authentication process. The main reasons that an IMSI is sent in the plaintext are because either a security context has not been established between the subscriber and the network yet, or the association of the subscriber and the corresponding security context has been lost. A security context is established between two entities, once they generated and agreed upon a shared secret key to further scramble data and signaling communications. This protects the confidentiality of the communications. However, before the establishment of the security context, the messages are not confidentiality protected in LTE. Here, sending the IMSI in plaintext makes sniffing this identity easy for an eavesdropper. Possessing the IMSI by an attacker can also lead to other attacks like impersonation and DoS.

Comparing the threat on the IMSI with the other threats (i.e., the location tracking and paging attack) shows that the latter two are much easier to address. First, the corresponding attacks occur after a security context has been established between the subscriber and the network. In this respect, preventing such attacks in an efficient way is a straight forward task. For instance, encrypting the signalling messages using a symmetric encryption scheme like AES can prevent the attacks. Second, those two attacks are performed on temporary identities. Such identities should be changed with the movements of the subscriber from one cell to another or from one location area to another. Here, to prevent the attacks, a randomizing technique can be utilized in the process of generating the temporary identities. This technique hides the relationship of the two consecutive temporary identities. For example, the sequence number used in the generation of the C-RNTI can be randomized which prevents the location tracking threat.

The above vulnerabilities, in general, are considered more critical in LTE-based PSBNs than in LTE-based commercial networks [11, 8, 128]. This is because an emergency is a critical situation in which human lives might be in danger. Here, even a trivial vulnerability in the security architecture of LTE or an attack on the network subscribers may have devastating consequences. For example, a successful location tracking of a high-ranked ER (e.g., a battalion chief) along with some other groups of responders who are all moving toward the same location area can be used as an indication of a more specific emergency response behaviour of the public safety personnel. Considering the preceding scenario, a group of adversaries can intentionally produce an emergency to mobilize the ERs to another location as a distraction to be able to successfully launch their attack in another location area. Those adversaries can use location tracking to ensure that their distraction has depleted a sufficient number of emergency response resources.

To prevent the subscriber permanent identity vulnerability, the previous works that were reviewed in Chapter 2 mainly proposed to use public-key encryption. Observe that public-key encryption is a natural way of addressing this problem since a security context has not been established between the subscriber and the network. Applying public-key encryption, a subscriber encrypts his/her IMSI using the public-key of the HSS which is the

entity who authenticates the subscriber. The public-key of the HSS can be stored/retrieved at/from a public repository. Then, the encrypted message is transferred for authentication. In this case, an adversary cannot sniff the IMSI and no security context is required. In terms of efficiency, however, public-key encryption causes high delay and increases computation and communication overheads. This makes it an inefficient solution.

In this chapter, we propose two algorithms to protect the confidentiality of an IMSI. The first one conceals an IMSI within a random string of bits to protect the confidentiality of the identity and privacy of the ERs. In the security analysis, we heuristically show that achieving full security for this algorithm is infeasible and makes the algorithm very inefficient. However, we leverage the ideas of the first algorithm to propose our second algorithm. The second algorithm applies those ideas along with the AES symmetric encryption scheme. The security analysis of the algorithms shows that the second algorithm is fully secure and feasible. HMAC (with SHA-256) has been used in the construction of the two algorithms. We considered 128 bits security for AES and HMAC. HMAC and AES impose light-weight computation complexity in comparison with the public-key cryptographic algorithms such as RSA. The performance analysis of the second algorithm shows its efficiency and effectiveness in comparison with public-key cryptography.

The remaining sections are as follows. Section 3.2 elaborates the system model and threat model of the proposed algorithms. Section 3.3.1 presents the proposed algorithms that prevent subscriber permanent identity threat. The security analysis of the algorithm is discussed in section 3.4. Performance evaluation of the algorithm is presented in Section 3.5. Section 3.6 presents our concluding remarks.

3.2 System Model and Threat Model

In PSBNs, ERs are going to use smart devices (that we call Smart Portable Devices (SPDs)) with high-performance capability and memory storage availability [128, 3]. For example, a smartphone or proprietary communication device can be used as an SPD [129]. Note that SPDs are compatible with the LTE standard. In addition, the public safety personnel (e.g.,

police, firefighters, and medical emergency personnel) are distributed into several public safety stations in different location areas around a city. In this regard, each ER belongs to a specific station that is located in a particular location area in a city. It is assumed that every location area may contain several distinct public safety stations.

At the beginning of an ER's shift, the station authorities authenticate the ER. This authentication process is out of the scope of this manuscript. After the authentication, the ER receives the gear needed for his/her shift including an SPD. The SPD is personalized for the corresponding ER at the beginning of his/her shift [128]. Note that SPDs are shared among the ERs in the corresponding station. Personalizing an SPD is done so that any message to/from the ER is marked with his/her unique identity. The main reason for that is to accomplish accountability if required [128]. This identity is different than the IMSI. An IMSI points to an SPD, but the aforementioned unique identity points directly to an ER.

It is worth mentioning that since the SPDs are shared among the ERs in a station, sniffing an IMSI may not directly point to a particular ER. Note that such an attack (i.e., mapping an IMSI to a particular ER) can still be accomplished if an attacker can have access to the log information that specifies which SPD is used by the ER during a specified time interval. Regardless of the preceding more sophisticated attack, sniffing an IMSI can still point to a particular location area and a public safety station. Here, during an SPD personalization, the SPD tries to connect to the network for which the attach request is sent to HSS. This transaction suffices for an attacker to be able to point an IMSI to a public safety station and the corresponding location area. This can lead to several other attacks as mentioned in Section 3.1.

It is assumed that the personalization process also includes preloading an array of secret keys and an array of initial patterns. The preloading process is done in a secured fashion. In other words, it is assumed that an attacker cannot sniff the two arrays during the preloading process. This setting (i.e., the preloading phase) is hard to achieve in a commercial network as the number of subscribers is large. However, in PSBNs, because the number of subscribers is much smaller than commercial networks and the SPDs are

personalized at the beginning of an ER's shift, the preloading process is achievable. An initial pattern is a random bit string of an arbitrary length more than 80 bits. A secret key is going to be used for the computation of the HMAC and AES algorithms and is considered to have 128-bit length. The two arrays are generated by HSS and distributed to the public safety stations.

It is assumed that all of the stations located in the same location area receive the same secret keys and initial patterns. On the other hand, the stations that are located in different location areas receive different arrays of initial patterns and secret keys. Here, HSS stores all of the arrays for all of the location areas under its jurisdiction. In order to protect the secret keys and initial patterns during storage, the arrays are securely stored in SPDs and HSS. In addition, it is natural to assume that HSS is a trusted entity. In addition, unauthorized individuals (i.e., the non-public safety personnel and the public safety personnel assigned to different location areas) should not be able to sniff an IMSI or correspond an IMSI to a location area or a public safety station during the exchanges of identity request and response messages in LTE.

3.3 Algorithms to Enhance Security and Privacy of LTE

In this section, we propose two algorithms to prevent the subscriber's permanent identity threat.

3.3.1 Hidden Identity Algorithm 1 (HIA1)

To prevent the subscribers' permanent identity threat, we propose an efficient algorithm Hidden Identity Algorithm 1 (HIA1). The idea of HIA1 is to conceal an IMSI within a random looking bit-string of a certain size. The size of the bit-string depends on the security of the algorithm. We will heuristically evaluate it in our security analysis in Section 3.4.

The bit string can be generated before or right after an ER receives an identity request message from an MME. An MME is an entity in the LTE between subscribers and HSS which provides services such as handover for the subscribers in a certain location area. When a subscriber sends an attach request to the network, the MME that covers the subscriber's area initiates the authentication process by sending back the identity request message to the subscriber. In response, the subscriber transfers his/her IMSI to the MME. To complete the authentication, the MME transfers the IMSI to HSS for further processing. In fact, LTE performs a mutual authentication process enabling both a subscriber and the network to authenticate each other. This mutual authentication process is out of the scope of this work.

An IMSI consists of three parts: MCC, MNC, and Mobile Subscription Identification Number (MSIN). MCC is three digits long, and MNC is two or three digits long. The longest part of an IMSI is the MSIN which is up to 10 digits long (i.e., approximately 40 bits). An IMSI is presented in (3.1) where || illustrates concatenation.

$$IMSI = MCC||MNC||MSIN. \tag{3.1}$$

Observe that in a country, MCC is the same for all of the IMSIs. In addition, an MNC represents a network provider in the country which can be varied. However, since the subscribers of PSBNs are all ERs, we can assume that they all operate over the same network. Therefore, the MNCs parts of different IMSIs in a country are going to be the same. Even if they are not the same, a small number of network providers support all of the ERs in a country. Consequently, it is the MSIN part of an IMSI that is changing from one ER (i.e., a subscriber) to another and this is the part that brings about the uniqueness for an IMSI. Therefore, instead of concealing the entire IMSI, we only consider MSIN.

To generate the random bit-string, an initial pattern P_s is used in concatenation with a timestamp T_c and a nonce n_c as the inputs to the HMAC function (3.2). A nonce is a random number that is generated by a random number generator and is changed for every HMAC computation. The nonce and timestamp are used to prevent the replay attack. HMAC is a keyed hash function; thus, we choose one of the secret keys in the secret key

array at random to compute the HMAC. The algorithm also retrieves the array indexes of the chosen initial pattern and secret key.

$$H_{K_s,P_s} = HMAC_{K_s}(P_s||n_c||T_c). \tag{3.2}$$

After computing the hashed value (i.e., H_{K_s,P_s}), we will conceal the MSIN of an IMSI within the hashed value. To do that, the bits of the hashed value are substituted with the bits of the MSIN in a way that the MSIN could be extracted by HSS. Here, the ones and zeros of the hashed value are going to be substituted at random by the zeros and ones in the MSIN respectively. Recall that an MSIN is 34 bits long. This means that 34 bits of the hashed value should be substituted with the bits in the MSIN. Table 3.1 shows the pseudo-code of HIA1 where M_{ir} is the identity response message.

Table 3.1: HIA1 pseudo-code

Input:Secret key and initial pattern arrays and MSIN
Output: M_{ir}
1. Randomly select a K_s and P_s
2. Generate n_c and retrieve T_c
3. Compute H_{K_s,P_s}
4. Call $Subs(H_{K_s,P_s}, MSIN)$
4.1 returns $SHMAC$
5. Generate the M_{ir} and send to MME

A secure random number generator is used to generate 34 distinct random numbers in the range of the length of the hashed value. For example, suppose that the length of the hashed value is 256-bits, then the random number generator will generate 34 distinct random numbers within [0, 256). These numbers show the bit indexes in the hashed value and will be stored in an array in the ascending order. We call this array MSIN Substitution Array (MSA). We convert the hashed value to an array in a way that the highest bit in the hashed value is stored in the lowest array index

The first element in the MSA represents a bit index in the hashed value in which the corresponding bit will be substituted with the highest order bit of the MSIN. The preceding

mapping procedure is repeated for the rest of the elements in the MSA. However, the algorithm will only substitute ones with zeros and zeros with ones. Here, if the value of a certain bit in MSIN is, for example, one and the randomly selected bit in the hashed value is also one, a right hand shift is performed till a zero bit is found in the hashed value. At this point, the zero bit is substituted with the one. A similar procedure is followed to substitute a one bit in the hashed value with a zero bit in the MSIN. Tables 3.2 and 3.3 present the substitution and right hand shift pseudocode. Figure 3.1 shows the substitution process.

Table 3.2: Substitution subroutine, $Subs(X, Y)$

Input: $X = H_{K_s,P_s}, Y = MSIN$
Output: $SHMAC$
1. Instantiate an empty array $MSA[\ ]$ of size 34
2. for $0 \leq i < 34$
$MSA[i] = random.nextInt(\|H_{K_s,P_s}\|)$
3. Sort MSA in the ascending order
4. $j = 33$
/*We convert the hashed value to an array in a way that the highest bit in the hashed value is stored in the lowest array index */
5. $SHMAC[\ ] = Convert - to - aray(H_{K_s,P_s})$
6. for $0 \leq i < 34$
if $SHMAC[MSA[i]] \neq MSIN[j]$ {
$SHMAC[MSA[i]] = MSIN[j]$ }
else if Right-hand-shift$(SHMAC, MSIN[j], MSA, i) == false$
Abort and call HIA1
j=j-1
7. return $Convert - to - bits(SHMAC[\ ])$

Notice that the right hand shift procedure in the substitution process will abort if the right hand shift reaches the next biggest random value in the MSA, before a substitution occurs. Here, the algorithm generates a new H_{K_s,P_s} using a new nonce and timestamp and proceeds with the substitution process.

Once the 34 random bits in H_{K_s,P_s} are successfully substituted with the bits in the

Table 3.3: Right-hand-shift $(SHMAC, MSIN[j], MSA, i)$

Input: $SHMAC, MSIN[j], MSA, i$
Output: Boolean value
1. $j' = 1$
2. while $\big(SHMAC[MSA[i] - j'] == MSIN[j]\big)$
if $(MSA[i] - j' \neq MSA[i+1])$
j'+=1
else return false
3. if $(MSA[i] - j' \neq MSA[i+1]\big)$
$SHMAC[MSA[i] - j'] = MSIN[j]$
return true
4. else return false

MSIN, the result is called Substituted HMAC ($SHMAC$). Then, the identity response message M_{ir} is formed as in (3.3) and sent to MME. In this equation, K_c is the secret key index in the secret key array and P_c is the initial pattern index in its corresponding array. The MME sends the message to HSS where the respective secret key and pattern will be retrieved using their respective indexes (i.e., K_c, P_c). Observe that the keys and initial patterns are generated by HSS. The HMAC will be calculated using the n_c, T_c, the initial pattern, and secret key. Afterward, the resultant hashed value will be compared with the SHMAC, and the difference is the MSIN.

$$M_{ir} = SHMAC||n_c||T_c||K_c||P_c \tag{3.3}$$

It is worth mentioning that considering the end-to-end delay (which is comprised of identity request and response transmission and the M_{ir} computation), the M_{ir} computation delay on the subscriber's side could be minimized to near zero. The subscriber could compute M_{ir} off-line, before it receives an identity request message. The only constraint

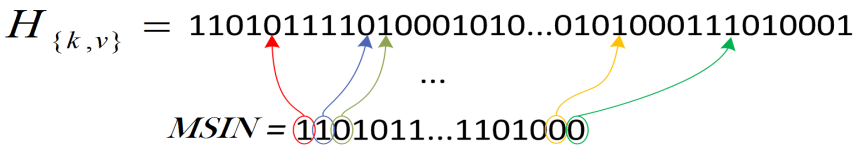

Figure 3.1: MSIN bit substitution process

is the timestamp which should be fresh. It is assumed that each SHMAC is valid for a specific time interval. So, the subscriber could calculate SHMAC right after a new time interval begins. Unless the beginning of a time interval does not overlap with the reception of an identity request message, a pre-calculated message can be sent in response to the identity request with zero delays.

Observe that, taking (3.3) into consideration, if an attacker is able to link two different M_{ir} messages to one another, the privacy the corresponding ER is violated. Linking the two messages is possible if either K_c or P_c of the messages are identical. In other words, if a subscriber uses the same secret key or initial pattern to generate two M_{ir} messages, the messages would have one element in common (i.e., K_c or P_c). This can imply that the two messages belong to one subscriber. Recall that the keys and initial patterns are shared among the ERs in the same location area. This means that even if two messages have one of the aforementioned elements in common, this does not necessarily mean that the messages belong to the same subscriber. However, this is still a viable threat since it can be exploited when integrated with other threats. For example, if two subscribers include the same secret key or initial pattern in the M_{ir} for the second time, the comparison of the signal strength of the two messages along with the comparison of the recorded first message's signal strength can locate a subscriber. Observe that the HIA1 algorithm also suffers from this point of vulnerability. In the following, we propose an algorithm to mitigate this vulnerability. This algorithm should also be employed for HIA1.

One of the ways that this vulnerability can be mitigated is to consider that the keys and initial patterns are only used once for a subscriber. This technique is cumbersome since HSS should regenerate the secret keys and initial patterns more frequently than the situation in which we can reuse them. Another way is to assume that the numbers of secret keys and initial patterns are large enough that the probability of selecting the same key or pattern for two different messages is negligible. This method requires a large memory size which may not be feasible for SPDs.

Aside from the above naive ways of tackling the problem, we propose a shuffling technique to permute the keys and initial patterns in a random way after using every initial

pattern and secret key in the corresponding array only once. For example, if there are $|S_{KP}|$ secret keys and initial patterns, a subscriber can generate $|S_{KP}|$ completely distinct M_{ir}s. Then, after the $|S_{KP}|^{th}$ message, the arrays of initial patterns and secret keys are randomly shuffled. This technique changes the indexes of the patterns and keys. Here, the shuffling is cryptographically protected so that an attacker cannot track the shuffles yet the synchronization of the subscribers and HSS would not be disturbed. We postpone the details of our proposed shuffling technique to Subsection 3.3.2.

3.3.2 HIA2

In order to protect the confidentiality of an IMSI in the identity response message against an eavesdropper, we can also use symmetric encryption algorithms such as AES. However, secret keys that are used to encrypt an IMSI should be shared between a subscriber and the HSS. Notice that the SPDs that ERs use should be personalized. Considering the ideas of the HIA1, it was assumed that the personalization phase includes preloading secret keys and initial patterns. Using the shared secret keys and initial patterns, we can encrypt an IMSI and send the corresponding ciphertext along with the indexes of the initial pattern and the secret key to HSS as shown in (3.4).

$$IMSI' = IMSI||P_s||T_c, \tag{3.4}$$

$$M_{ir} = E_{K_s}(IMSI')||K_c||P_c. \tag{3.5}$$

The IMSI has changed to $IMSI'$ so that the encryption of that also changes to a different value when the same secret key is used more than once. Although this technique protects the confidentiality of a subscriber's IMSI, identification and tracking of the subscriber are still possible which violate the privacy of the subscriber as mentioned in Subsection 3.3.1. Therefore, to further prevent additional threats, we propose a shuffling technique. Here, we modify (3.4) as follows,

$$IMSI' = IMSI||P_s||Counter_{SH}||T_c, \tag{3.6}$$

where $Counter_{SH}$ is the counter that keeps the number of times that the original arrays of keys and patterns have been shuffled. To maintain the synchronization between the subscriber and HSS, the last $IMSI'$ that is generated includes two counters as shown in (3.7).

$$IMSI' = IMSI||P_s||Counter_{SH}(old)||Counter_{SH}(new)||T_c. \tag{3.7}$$

Therefore, the final message is used to notify HSS that a shuffle is required. After HSS decrypts the M_{ir}, it extracts T_c and compares it to the current time interval. In addition, it extracts the counter. The counter points to specific secret keys and initial patterns arrays. Then, HSS checks if the P_c corresponds to the same P_s. If any of the preceding checks do not hold, HSS aborts the procedure.

To perform the shuffle, we use HMAC as illustrated in Table 3.4. Tables 3.4, 3.5, and 3.6 only illustrate the procedures for the secret key array. However, the same procedure is performed for the initial pattern array as well. In Table 3.4, first, we assume that HSS and all of the public safety stations (e.g., a fire station, police office, etc.) in the same location area share a long-term shuffling key $K_{shuffle}$. This key is also deployed to SPDs of ERs during the preloading process in a secured way. Second, the shuffling is done using (3.8) and (3.9),

$$K_c(new) = HMAC_{K_{shuffle}}(K_c(old)||T_c) \mod m_s, \tag{3.8}$$

$$P_c(new) = HMAC_{K_{shuffle}}(P_c(old)||T_c) \mod m_s, \tag{3.9}$$

where $K_c(new)$ is the new secret key array index. Here, the secret key that was located at the $K_c(old)$ index in the secret key array is now placed in the $K_c(new)$ index. The size

Table 3.4: Shuffling procedure

Input:$K_{shuffle}$, $K_S[Counter_{SH}][m_s]$, $P_S[Counter_{SH}][m_s]$, $K_c(old)$
Output:$K_S[Counter_{SH}+1][m_s]$, $P_S[Counter_{SH}+1][m_s]$
1. Boolean Shift = true
2. for $0 \leq i \leq m_s - 1$
Generate $K_c(new)$
if $K_S[Counter_{SH}+1][K_c(new)]$ is not empty
if $Shift == true$
$Shift = false$
Call RShift(Shift, $K_c(new)$, $K_S[Counter_{SH}+1][m_s]$,
$K_S[Counter_{SH}][K_c(old)]$, $K_c(old)$)
else
$Shift = true$
Call LShift(Shift, $K_c(new)$)
else $K_S[Counter_{SH}+1][K_c(new)] = K_S[Counter_{SH}][K_c(old)]$

Table 3.5: RShift

Input: Shift, $K_c(new)$, $K_S[Counter_{SH}+1][m_s]$, $K_S[Counter_{SH}][K_c(old)]$, $K_c(old)$
Output: Placing $K_S[Counter_{SH}][K_c(old)]$ to the right of $K_c(new)$
1. $j = 1$
2. while $(K_S[Counter_{SH}+1][K_c(new)+j])$ is not empty
if $(K_c(new)+j) == m_s + 1$
Call LShift(Shift, $K_c(new)$, $K_S[Counter_{SH}+1][m_s]$,
$K_S[Counter_{SH}][K_c(old)]$, $K_c(old)$)
$j++$
3. $K_S[Counter_{SH}+1][K_c(new)+j] = K_S[Counter_{SH}][K_c(old)]$
4. return

of both secret key and pattern arrays is $m_s = |S_{KP}|$ and T_c is a timestamp that is used to randomize the process. As illustrated in Table 3.4, if the array cell to which the new index points was already occupied, the algorithm performs a right-hand shift or a left-hand shift. The right or left-hand shifts are based on a boolean value *Shift*. If the *Shift* is *true* a right-hand shift subroutine is called, otherwise, a left-hand shift subroutine is called. This boolean value changes every time a shift has occurred. The shift processes continue until an empty cell is reached. Here, if there was no empty cell left at right or left-hand shift, the opposite direction with respect to the current shift is chosen. Tables 3.5 and 3.6 illustrate the shift procedure.

Table 3.6: LShift

Input: Shift, $K_c(new)$, $K_S[Counter_{SH}+1][m_s]$, $K_S[Counter_{SH}][K_c(old)]$, $K_c(old)$
Output: Placing $K_S[Counter_{SH}][K_c(old)]$ to the left of $K_c(new)$
1. $j = 1$
2. while ($K_S[Counter_{SH}+1][K_c(new)-j]$) is not empty
if ($K_c(new) - j == -1$)
Call RShift(Shift, $K_c(new)$, $K_S[Counter_{SH}+1][m_s]$,
$K_S[Counter_{SH}][K_c(old)]$, $K_c(old)$)
$j--$
3. $K_S[Counter_{SH}+1][K_c(new)-j] = K_S[Counter_{SH}][K_c(old)]$
4. return

3.4 Security Analysis

In this section, the security of the HIA1 and HIA2 protocols is discussed heuristically. In addition, the HIA2 algorithm is studied with respect to several attacks.

3.4.1 Security Analysis of HIA1

Recall that HIA1 substitutes the bits of a hashed value with the bits of an MSIN. For this to be secure, the resultant SHMAC should have the same distribution as the original HMAC. To model the HMAC hash function, we use the random oracle model [130, 131]. In general, this model represents a mathematical function that can be queried by anyone and maps every query to a uniformly and randomly chosen response from its output domain. In practice, random oracles can be used to model cryptographic hash functions. In addition, we use the random oracle to generate the 34-bit indexes that point to the locations of the substitutable bits. The random oracle can be represented as in (3.10) and (3.11),

$$H1 : \{0,1\}^* \xrightarrow{R} \{0,1\}^{l_{HMAC}} \tag{3.10}$$

$$H2 : \{0,1\}^* \xrightarrow{R} Rand \in [0, l_{HMAC}-1] \tag{3.11}$$

where H1 is used to obtain the hashed value, H2 is used to obtain the bit indexes, and

l_{HMAC} is the size of the hashed value in bits. To check if the distribution of the SHMAC is similar to a hashed value, we need to see how the hashed value is influenced by the bit substitutions. Here, we give an example to illustrate the situation.

Note that the number of zeros and ones in the HMAC result and the placement of the bits follows Binomial distribution as (3.12),

$$\Pr(X = k) = \binom{l_{HMAC}}{k} p_r^k (1 - p_r)^{(l_{HMAC} - k)}, \tag{3.12}$$

where k shows the number of zeros or ones. In (3.12), the highest probability is obtained when $k = l_{HMAC}/2$ which implies that the number of zeros and ones are the same in the hashed value. Suppose the preceding situation has occurred. In other words, suppose the number of zeros and ones are equal in the hashed value. In addition, suppose that the 34 queries to H2 result in 1 bit index every $x = \lceil l_{HMAC}/34 \rceil$ bits. For example, for $l_{HMAC} = 256$ bits, among every 8 bits of the hashed value, one will be substituted with one bit in the MSIN. The number of zeros and ones in those 8 bits also follow the Binomial distribution. Since the random oracle model produces the uniform distribution, then, $p_r = 0.5$.

In the aforementioned example, we can imagine that each possibility (i.e., the number of zeros and ones in every $x = 8$ bits) creates a state that might occur with a certain probability as shown in Figure 3.2. To substitute one of the 8 bits with a bit in MSIN, we simply make a transition from one state to a neighboring state. The states for zero bits and one bits in Figure 3.2 are shown to provide clarity since they complement each other. Basically, the two sets of states represent the same event. Assuming that the MSIN distribution is also uniformly random, then with equal probability we might transit to a neighboring state.

For the example shown in Figure 3.2, the state of the selected bits of the hashed value is shown with filled ellipses. Here, the number of zeros equals $x/2 - 1$ and the number of ones equals $x/2 + 1$. These two states occur with the same probability as in (3.13). Now, suppose we insert a 0 bit to this selection of bits. In this case, we make a transition to

the $x/2$ state for both zero bits and one bits (as shown by an arrow marked with a star). This way the probabilities of a zero bit or a one bit substitutions are equal. Therefore, the transition to the $x/2$ state gives no additional information to an adversary. On the other hand, if a one bit is inserted, we make a transition to $x/2-2$ for zero bits, and another one to $x/2+2$ for one bits. As the probability of these states are small before any transition occurs, an adversary can guess which transition we made with high probability.

$$\Pr(0\ bits = 3) = \binom{8}{3}0.5^3 0.5^5 = \Pr(1\ bits = 5) = \binom{8}{5}0.5^5 0.5^3 = 56/256, \tag{3.13}$$

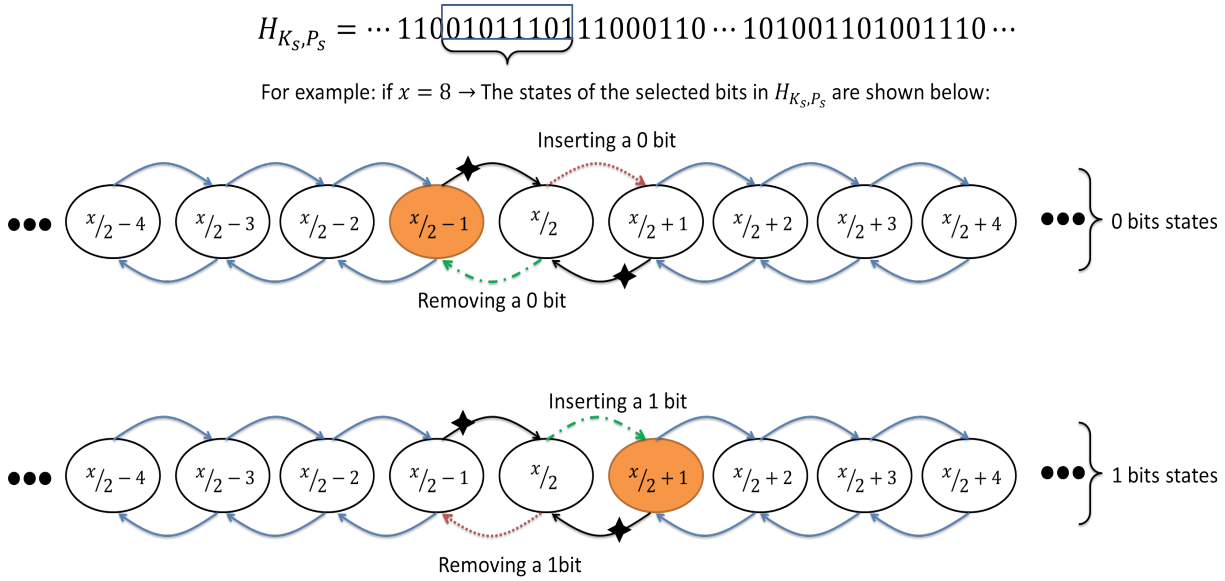

Figure 3.2: Transition states

It should be noted that achieving this state all the time may be infeasible. Here, a secured HIA1 faces two main boundaries: one, the distribution of MSIN is not necessarily uniform; two, a transition to the middle state (e.g., the $x/2$ state in the preceding example) is not easily achievable. If two or more bit indexes are very close to one another, and all of them substitute the same bit (zero or one), the SHMAC distribution will look different than a random bit string. To mitigate that, we need to increase the length of the hashed value. However, this makes the algorithm inefficient in terms of computation and communication overheads. Although it is hard to achieve a secured and yet efficient HIA1, our HIA2 is

secure, efficient, and feasible which will be discussed in the next subsection.

3.4.2 Security Analysis of HIA2

The security of the HIA2 algorithm mainly relies on the security of the symmetric encryption scheme. Since AES is thought to be proven secure, therefore, our proposed scheme can also be considered secure. However, the shuffling function should be analyzed to make sure that it is secure. In other words, for an adversary, linking the secret keys or initial patterns indexes before and after the shuffling function should be intractable as shown in (3.14) and (3.15).

$$Shuffle(K_c(old)) \rightarrow K_c(new) \tag{3.14}$$

$$Shuffle^{-1}(K_c(new)) \nrightarrow K_c(old) \tag{3.15}$$

We used HMAC in the shuffling function; however, unlike the HIA1 we do not need to rely on the random oracle model. Here, HMAC is considered as a cryptographic hash function for which it is intractable to invert the function (i.e., obtaining the input from its output). This property ensures that it is infeasible for an adversary to be able to link $K_c(new)$ to $K_c(old)$. In addition, we used the timestamp to randomize the HMAC. In this regard, shuffling an index twice produces different results as shown in (3.16),

$$HMAC_{K_{shuffle}}(K_c||T_c) \neq HMAC_{K_{shuffle}}(K_c||T_c'), \tag{3.16}$$

where K_c is an index in the secret key array, and T_c and T_c' are two different timestamps. This ensures that the hashed value of an index is varied every time that it is computed. Therefore, the preceding two properties that HMAC provides are sufficient for our shuffling function to be secure. However, it has to be mentioned that the shuffling key and the secret keys are stored in an SPD. This is still a potential point of vulnerability. This can be protected if the keys are encrypted and stored. In the following, we analyze HIA2 with

respect to several types of attacks.

Subscriber Anonymity

Subscriber anonymity is an integral part of privacy preservation. Using AES to encrypt an IMSI and considering our proposed secure shuffling technique, the subscriber anonymity is achieved.

Subscriber Untraceability

HIA2 eliminates any chance to identify the past identity requests and responses of the same subscriber. In other words, the attacker cannot determine which messages have been sent from a single subscriber. The untraceability is provided because the algorithm uses a timestamp as an input to the AES and HMAC. Thus, each time the algorithm is executed the result would seem random and different from previous ones. Moreover, the initial pattern and key are chosen at random for every execution of the algorithm. This eliminates any order that might be inferred from the messages.

Subscriber Unlinkability

Between two consecutive shuffling processes, the secret keys and initial patterns are used only once. This ensures that the indexes included in the M_{ir} are not repeated before the arrays are shuffled.

MME Impersonation Attack

Suppose that an adversary pretends to be an MME and sends an identity request message to some random subscriber in a particular location area. Upon receiving the identity response, the adversary will not be able to obtain the IMSI and identify the subscriber. This is because the illegitimate MME does not have access to the patterns and keys. Furthermore, because of the mutual authentication procedure between subscribers and

HSS in LTE networks, the MME impersonation attack will be prevented. Moreover, in PSBNs, based on [132], the existence of a trusted HSS is important so as to avoid any Sybil attack.

Replay Attack

Using a timestamp prevents the replay attack. Each message has a short period of validity, which reduces the probability of any repeated messages sent by an attacker.

Subscriber Impersonation Attack

The same situation as the MME impersonation applies here. The HSS will easily detect fake messages of an unauthorized subscriber by the mutual authentication procedure.

3.5 Performance Evaluation

The proposed HIA1 and HIA2 algorithms have been implemented using the Eclipse environment and the Java language. The simulations have taken place on a desktop PC using a 64-bit Windows 7 operating system. The PC is running on a Core i3 CPU with a processing speed of 3.3 GHz. In our simulations, the delay corresponding to the computations at the HSS was considered. The computation delays of the HIA1 and HIA2 algorithms on the subscriber side can be neglected since this process can be done before an identity request is received (i.e., off-line). Considering a 95% confidence interval (see Appendix A for further information), the simulation results illustrate the computation delay of our algorithms in comparison with public key encryption algorithms RSA and Elliptic Curve (EC) ElGamal.

The computation delays of HIA1, HIA2, RSA, and EC-ElGamal are compared in Figure 3.3. In our simulations, the key sizes of RSA and EC-ElGamal algorithms were chosen to be 2048 bits and 224 bits respectively [133]. Figure 3.3 shows the difference in those algorithms when the number of simultaneous identity responses increases (i.e., number of

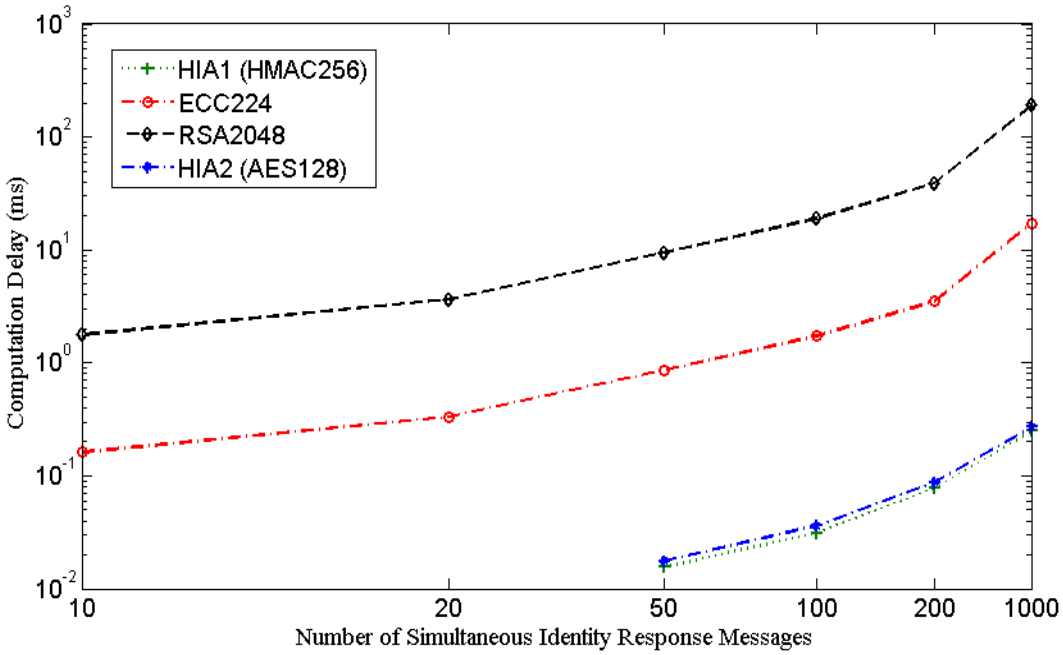

Figure 3.3: Identification delay comparison

simultaneous computations of the respective algorithms). The results indicate that when the number of simultaneous identity responses rises, the computation delay of RSA and EC-ElGamal increases significantly in comparison with the HIA1 and HIA2 protocols. In this case, for 100 simultaneous identity response messages, RSA is 11 times slower than EC-ElGamal, and approximately 300 times slower than HIA1 and HIA2. Observe that the computation delay of both HIA1 and HIA2 are negligible when the number of simultaneous identity responses is less than 50.

Table 3.7 summarizes several protocols that address the permanent identity threat based on the communication overhead, computation complexity, and memory management computations. As is illustrated, the communication overhead of HIA2 is lower than [17, 52, 53]. Compared with [52, 53], HIA2 uses AES which has a 128-bit block size while the other two have 2048 or 224 bit moduli with respect to the chosen public key scheme (e.g., RSA or EC-ElGamal respectively). The lengths of K_c and P_c are selected to be 16 bits as this amount can provide the maximum number of required keys and patterns. Considering the preceding overhead items, the communication overhead of HIA2 in comparison with RSA and EC-ElGamal is approximately 12 and 1.5 times lower respectively. And,

Table 3.7: Protocol Comparisons

Scheme	Solution	Communication Overhead (bits)	Computation Complexity	Memory Management Computation	Memory (bits)
[17]	SHA160	$2 \times 128 + 16$	$4(SHA160) + AES_{128}$	$4O(t_{reg}) \times (SHA160)$	4×128
[52, 53]	RSA, EC-ElGamal	2048, 224	RSA_{2048}, ECC_{224}	-	2048, 224
HIA2	AES and HMAC-SHA160	$128 + 32$	$AES_{128} + 1(HMAC - SHA160)$	$2 \times O(m_s)(HMAC - SHA160)$	$2^{\lvert K_c \rvert + 1} \times m_s \times 128$

the communication overhead of HIA2 in comparison with [17] is approximately 1.7 times lower. Similarly, the computation complexity of HIA2 is lower than [17] by 3 hash function computations.

Our proposed HIA2 and [17] perform memory management for the secret keys and initial patterns (e.g., the shuffling algorithm in HIA2). The amount of necessary memory management effort in [17] is proportional to the number of registered subscribers t_{reg}. For the memory management computations, we neglected the computation complexities of shift procedures and memory access since the hash function computations are dominant. As the number of subscribers increases, the required effort increases linearly. On the contrary, the memory management in our scheme depends on the number of secret keys and initial patterns. Since the proposed shuffling algorithm appears to be secure, those numbers can be kept constant. In addition, the memory management in [17] is a continuous process while in HIA2 it can be done in bursts and off-line. Both of the algorithms are considered to be applicable for emergency situations (i.e., non-delay-tolerant applications) since their computation complexities at HSS are light-weight, thus the corresponding delays are small. However, if the number of subscribers increases, memory management of [17] may overwhelm HSS and the applicability of the algorithm may be changed to delay tolerant applications.

Concerning the memory requirements, as Table 3.7 indicates, HIA2 requires more memory than the rest of the algorithms. For $|K_c| = 16$ bits (or $|P_c| = 16$ bits), the number of secret keys or initial patterns that are stored will be $2^|K_c|$, and the length of a secret key or an initial pattern is 128 bits. However, [17] only stores 4 RICs at a time. Observe that since the shuffling function appears to be secure, we can reduce the $|K_c|$ (or $|P_c|$). Here,

there is a trade-off between the shuffling computations and the required number of secret keys and initial patterns.

3.6 Conclusion

In this chapter, we investigated the main security and privacy vulnerability of the LTE which is the permanent identity threat. To tackle the threat, we proposed two algorithms HIA1 and HIA2. The security analysis of the two algorithms showed that HIA2 is superior to HIA1. The HIA2 algorithm also prevents replay attack, MME and subscriber impersonation, and provides subscriber untraceability and unlinkability. In addition, we provided the performance analysis of the proposed algorithms compared with RSA and ECC public-key cryptography. The performance analysis showed that HIA2 is more efficient in terms of computational and communicational overheads than the RSA and EC-ElGamal algorithms and [17].

Chapter 4

Context and Location-Aware Data Availability Scheme

4.1 Introduction

In this chapter, we focus on privacy-preserving data availability in the context of an emergency. Suppose a large building (e.g., with 10 floors) is on fire and many people are trapped inside, or a region with a population of 1000 civilians and 300 houses faces a natural incident like an earthquake or flood. Under such circumstances, ERs seek all possible information to achieve SA. In general, they would like to find answers to the following questions: How many people are in danger? Is there a way to identify those individuals? What is the closest location of the endangered individuals? What were the health conditions of people before the incident? What are their health conditions at the moment? Where can we find the health records of endangered individuals? Are there people aside from ERs who are close to the situation and who have certain capabilities that can be used to help others, like engineers, physicians, nurses, etc? Many previous incidents have shown that people care about each other in hard times and, in fact, volunteer to help others in need. In this case, another question would be, how is it possible to reach out to those available volunteers in critical situations and ask them for their help?

The answers to the aforementioned questions can profoundly help ERs in emergency operations. However, there are substantial challenges towards designing a system which addresses such requirements. First, emergency situations are highly dynamic. This mandates strict requirements including high scalability, data availability, and very low response time. This is because in such cases a large number of people may be affected, among which many may require immediate care. Second, the sought information is vastly distributed which makes the data retrieval process even more complex. For example, Physical Health Records (PHRs) are stored in proprietary hospitals, or in various cloud servers like Amazon, Google, and Microsoft to name a few. In fact, before retrieving any Personal Information (PI) like PHRs, it is necessary to identify endangered individuals and the servers to which they have outsourced their information. Without proper identification, no information can be retrieved. Third, the sought information is considered Personally Identifiable Information (PII) which raises privacy issues.

In this chapter, we propose a scheme to answer the aforementioned questions raised above and to tackle the issues highlighted in Chapters 1 and 2 regarding data availability. Our scheme not only provides a sufficient level of SA for ERs, but also it addresses the prerequisite step for data retrieval which is privacy-preserving user and server identification. In this regard, we propose a secure data storage structure to store "meta" PI. Meta PI is merely comprised of keywords (e.g., a health condition like asthma, the permanent location, identity of the server storing the complete PI, etc.) that describe the specific conditions of a DO. To provide high data availability, we utilize the cloud storage model and an opportunistic storage model to store our data structure in a central cloud server and several distributed mobile storage units. We propose a privacy-preserving search algorithm to facilitate data retrieval. The scheme also offers multi-keyword search for conjunctive and disjunctive queries. The search algorithm imposes minimum delay which is desirable for emergency situations. We provide extensive security analysis, simulation studies, and performance comparison with the state-of-the-art solutions to demonstrate the efficiency and effectiveness of the proposed approach.

The remaining sections are as follows. Our system model, threat model, and assumptions are presented in Section 4.2. Section 4.3 elaborates our scheme construction. Security analysis and performance evaluation are discussed in Sections 4.4 and 4.5 respectively. Section 4.6 summarizes the chapter.

4.2 System model and Threat model

It is assumed that a city is divided into several distinct location areas, each having a unique pseudo-identity PS_a. We assume that one Central Cloud Server (CCS) stores all data. The system is comprised of several entities as follows.

Key Generation Authority (KGA): This entity generates the secret keys of the system.

Cloud Servers (CS): In addition to CCS, we assume that there are several cloud servers, each managed by a different vendor such as Google, Microsoft, Amazon, and so forth. These servers store the complete version of PI for individuals which may be up to 200 pages per record [134].

DO: This entity is a member of the general public. People upload their encrypted data such as PHRs to central and mobile clouds.

ERs: these are the governmental authorities including police officers, firefighters, and paramedics.

Mobile Cloud (MC): we assume that there are individuals who possess powerful smartphones with high computing capabilities and additional storage. Other individuals in their local proximity will upload their data to these providers either before an incident or during one.

Adversary: This entity will try to eavesdrop on the communications and send queries to CCS to retrieve information. This work does not focus on DoS.

As depicted in Figure 4.1, there is an area in which an emergency situation has occurred. Note that before an incident happens and during normal conditions, those DOs who have

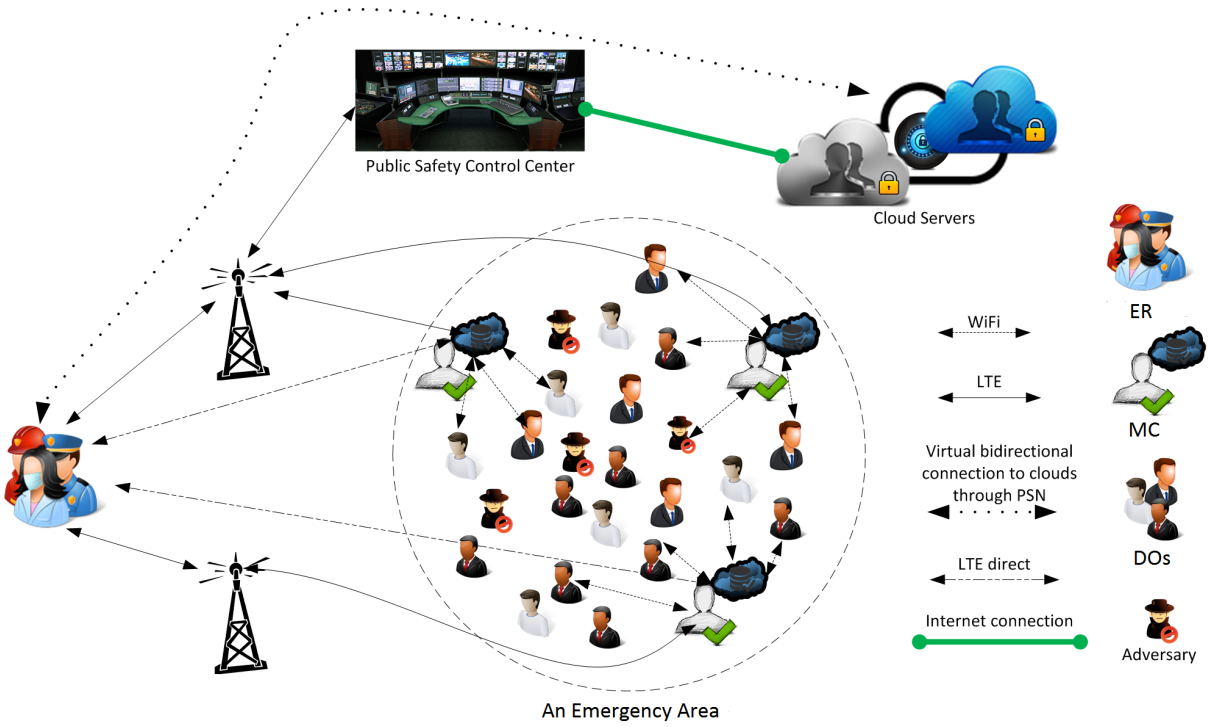

Figure 4.1: System model

registered to the system will outsource their encrypted Meta Information (MI) using our algorithm to the CCS and one or many MC(s). MI is comprised of a health attribute keyword set (e.g., $HA_u = \{Asthma, A^+, etc.\}$), a pseudonym of the DO $U_{P_{ID}}$, the identity of a CS where a complete version of their PHR is stored (e.g., CS_x), the memory index σ_u where the PHR is recorded in the CS, and other kinds of PI relevant to an emergency PI_{em} (e.g., emergency contact number, qualifications (e.g., civil engineer, electrician), etc.). It has the following format, $MI = \{U_{P_{ID}}||HA_u||CS_x||\sigma_u||PI_{em}\}$. This information is generated by DOs and uploaded to the CCS or an MC. In an emergency situation, ERs can search for disjunctive/conjunctive health attributes and PI_{em} keywords for a particular affected area, and retrieve such information.

In this chapter, we assume that KGA is fully trusted, but CSs and MCs are honest but curious. This means that they follow the procedure of the scheme in an honest way, but try to learn as much information as possible. We assume that KGA authenticates DOs and ERs and only then it transfers secret keys to those entities. However, the authentication procedure is out of the scope of this work. We also assume that there exist eavesdropper

adversaries who would like to learn as much information as possible.

4.3 Location-Aware Data Availability Scheme

4.3.1 Storage Bloom Filter

A Bloom Filter (BF) is a type of data structure that represents a set of l keywords by an array of m bits [135]. The inputs are as follows:

1. A set of l keywords: $W = \{W_1, W_2, ..., W_l\}$
2. An independent set of hash functions:$\{H_i\}_{i=1}^{r}$ where,

$$H_i : \{0,1\}^* \rightarrow [1, m] \text{ for } 1 \leq i \leq r. \tag{4.1}$$

3. A bit array of size m which is set to 0 initially.

For each element $W_j \in W$, the bits in m at positions $H_i(W_j)$ are set to 1 for $i = 1, ..., r$. To check whether a word W_w is in the BF or not, we may check $H_i(S)$ for $i = 1, ..., r$; if all of the resultant bits are 1, then W_w was included in the BF with high probability. Otherwise, even if only one position is 0, definitely W_w was not inserted.

BFs have an important feature called false positive rate f_p which means that a word W_w has not been inserted in the filter but in the test process, respective bits' locations $H_i(W_w)$, $\forall i$, are all 1. To quantify f_p, suppose we use r hash functions to insert l distinct keywords into an array of size m; the probability that bit j in the array is 0 is $(1-(1/m))^{rl} \approx e^{-rl/m}$. Thus, the probability of a false positive is approximately $(1-e^{-rl/m})^r$ [136]. It is reasonable to assume that the parameters m, l are fixed. So, to calculate the minimum false positive rate we will compute when the derivative of the probability of false positive rate (with respect to r) is equal to zero. Then, the minimum value equals $r = ((m/l) \ln 2)$, with false positive rate of $(1/2)^r$. Based on a desired f_p, the optimum relationship between BF parameters is presented in equation 4.2 [136]:

$$m = \frac{lr}{\ln 2}. \tag{4.2}$$

We modify a variation of BF called counting BF (CBF) to achieve Storage BF (SBF). CBF is a variation of the standard BF in which each bit of the BF acts as a counter [135]. Note that in a regular BF; any bit can be targeted more than once during the indexing process. But, once it has flipped to 1, its value does not increment further. However, for a CBF, the value of a counter increments every time that the counter was targeted. Now, to build an SBF, we assume that instead of counters in CBF, we have a set of buffers $B = \{B_1, B_2, ..., B_m\}$ to store data. Therefore, to insert a pair (W_i, Vw_i), $\forall W_i \in W$ into an SBF, Vw_i is added to the $B_{H_j(W_i)}$ for $j = 1, ..., r$ where $B_{H_j(W_i)}$ is the targeted buffer. Then, to check whether $W_i' \in B$ or not, one can check all the sets $B_{H_j(W_i')}$ and if all the sets are non-empty it returns the value associated with W_i' which is the intersection of all $B_{H_j(W_i')}$ (i.e., $\bigcap_{j=1}^{r} B_{H_j(W_i')}$).

4.3.2 Construction of the Scheme

Setup(S_{sec}): Given the security parameter S_{sec}, choose a pseudo-random function $f : \{0,1\}^n \times \{0,1\}^{S_{sec}} \rightarrow \{0,1\}^{S_{sec}}$ and a fixed set of keywords, (e.g., $W =$*{Asthma, Heart condition, artificial leg, blood type, civil engineer, electrician, etc.}* where $|W| = l$ and $W_i \in \{0,1\}^n$. For each W_i, KGA generates a secret key $\varsigma_i \in \{0,1\}^{S_{sec}}$, $\forall i \in W$. KGA also generates a public-private key pair for ERs considering the same security parameter S_{sec}.

In addition, a set of initial vectors $V = (v_1, v_2, ..., v_r) \in \{0,1\}^{nr}$ will be produced. Each user also has a CBF (CBF_u), and a standard BF (BF_u), both of which will be initialized to 0. On the server side, we utilize an SBF to store DOs' data. It is assumed that all buffers of an SBF are of the same size. ERs will receive the key sets with respect to their authorization.

Registration(HA_u): Upon a DO's registration and based on his/her set of keywords $W_u = \{HA_u \bigcup PI_{em}\}$, KGA will transfer a set of master keys $\varsigma = \{\varsigma_j\}$ for $1 \leq j \leq |W_u|$, along with V to the DO. Then, KGA generates the corresponding master keys as in (4.3)

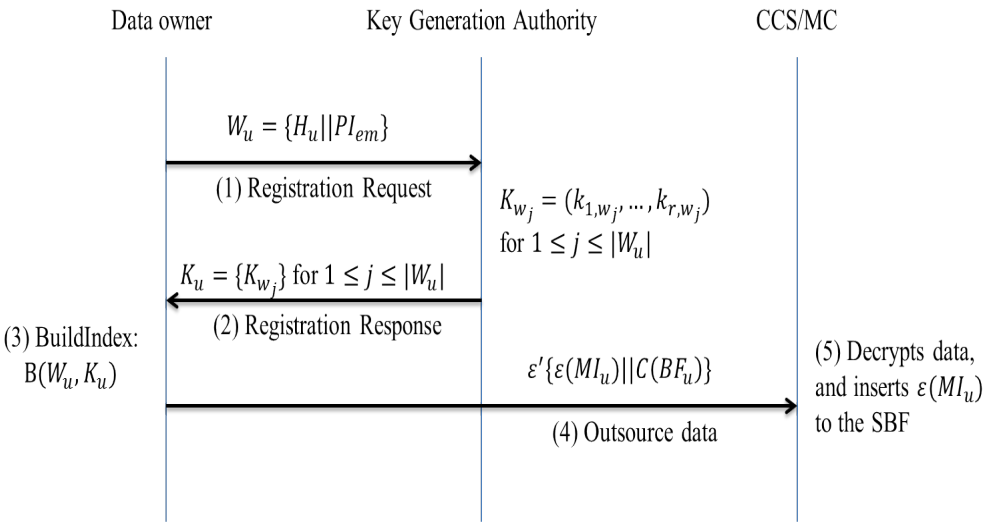

Figure 4.2: Communication paradigm between a DO and a CCS/MC

for all $W_j \in W_u$. Note that a master key is in the form of $K_{W_j} = (k_{1,W_j}, ..., k_{r,W_j})$. It is assumed that t_{reg} users will sign up to the system.

$$K_{W_j} = (f(v_1, \varsigma_j), f(v_2, \varsigma_j), ..., f(v_r, \varsigma_j)) \in \{0,1\}^{rS_{sec}}. \tag{4.3}$$

The communication paradigm between a DO and the KGA and CCS (or an MC) is illustrated in Figure 4.2. In addition, Figure 4.3 shows the communication paradigm between an ER and the KGA and CCS (or an MC). In both figures, the messages one and two show the registration request and response messages for the respective member of the system.

BuildIndex(W_u, K_u)**:** The input includes the set W_u as keywords and their corresponding master keys. The outputs are CBF_u, BF_u, and OBF_b. In Figure 4.2, the BuildIndex algorithm is shown as the third step of the construction of the algorithm. Figure 4.4 illustrates a high-level overview of the BuildIndex algorithm.

Step 1: For every $W_j \in W_u$, compute the following:

(a) Trapdoor: $T_{W_j} = \{z_1, z_2, ..., z_r\} \in \{0,1\}^{rS_{sec}}$ is calculated using (4.4),

$$T_{W_j} = \{f(W_j, k_{1,W_j}), ..., f(W_j, k_{r,W_j})\}. \tag{4.4}$$

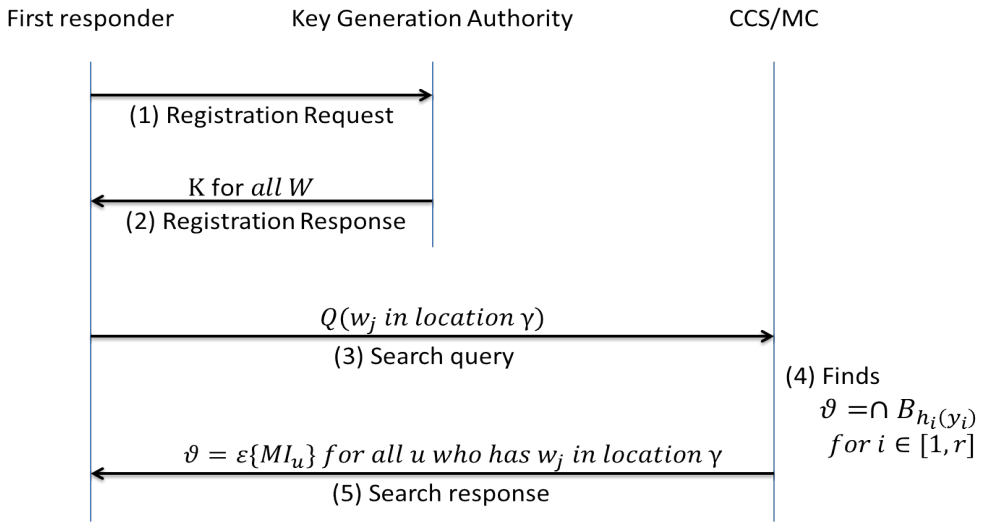

Figure 4.3: Communication paradigm between an ER and a CCS/MC

(b) Location vector: if $\gamma \in \{0,1\}^n$ is a specific location inside PS_a, then the location vector L_{W_i} is computed as in (4.5),

$$L_{W_i} = \{f(\gamma, z_1), ..., f(\gamma, z_r)\} = \{y_1, ..., y_r\} \in \{0,1\}^{rS_{sec}}. \tag{4.5}$$

(c) Insert $y_1, ..., y_r$ to both BF_u and CBF_u as follows:

$$\begin{cases} BF_u : & b_{H_i(y_i)} = 1, \\ CBF_u : & C_{H_i(y_i)} = C_{H_i(y_i)} + 1. \end{cases} \tag{4.6}$$

Where $i = 1, ..., r$. Note that $b_{H_i(y_i)}$ and $C_{H_i(y_i)}$ are bit location and counter location in BF_u and CBF_u respectively. And, BF_u and CBF_u both have the same length.

Step 2: We will build an Obfuscating BF (OBF) for some extra Obfuscating Elements (OEs) that are used to obfuscate BF_u. Suppose $max\{|W_u|\} = q < l$. For a DO, $|W_u| \leq q$ for which he builds (4.6). Then, the DO picks $(q - |W_u|)$ random values in $\{0,1\}^*$, computes (4.6), and only keeps BF_u which we name OBF. The DO will update BF_u by the bitwise OR operation of the two (i.e. $BF_u = BF_u \vee OBF$ where $\vee$ represents the bitwise OR operation). This way, every DO will have the same number of elements to add to the SBF.

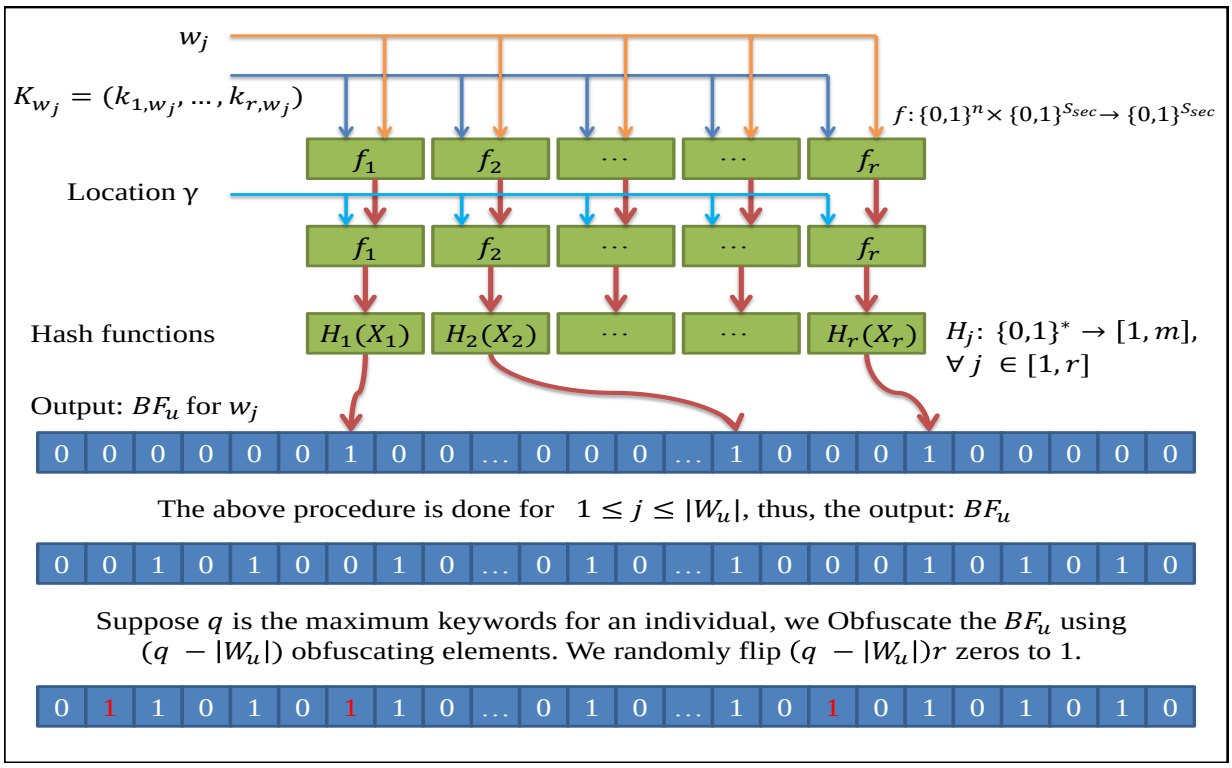

Figure 4.4: BuildIndex algorithm

Step 3: The output of the previous step is $I_U = (CBF_u, BF_u, OBF)$. Before the DO outsources data to a server, he/she compresses the BF_u to decrease communication overhead. Using data compression tools, one can significantly decrease the size of the BF_u to be sent. Thus, the DO calculates a compression function with BF_u as an input, $C(BF_u)$. The compression function is used because in a BF_u, the number of unchanged elements (i.e., that have a zero value) is much larger than the ones that have changed to 1 which is shown by λ in (4.7). In other words, λ is the expected number of distinct ones after indexing q items into the BF_u. If q is large, λ will be less than rq because some elements will overlap.

$$\lambda \simeq m - m \times e^{-rq/m} \leq rq \tag{4.7}$$

An example of a compression function is to consider that instead of sending the entire bit string, only send the bit indexes where a bit is one. For example, in a bit string of size 500, if the 50^{th} bit is the only 1 and others are zero, then sending the value fifty using six

bits would show such a bit index. This compression function is only efficient in terms of communication overhead if $|C(BF_u)| < |BF_u|$ in bits. Proposing a compression function is out of the scope of this thesis.

Then, the DO sends a packet to the CCS/MCs in the form of $\xi'\{\xi(MI_u)||C(BF_u)\}$ where $||$ means concatenation. The server simply decrypts the packet, decompresses $C(BF_u)$, and adds $\xi(MI_u)$ to the corresponding buffers in the SBF using BF_u. Note that using the ERs' public key and a secure public-key encryption scheme, we generate $\xi(MI_u)$. Also, note that it is assumed that a DO and a CCS/MC use SSL to secure communications between one another.

SearchIndex(W_i, γ)**:** The search query and response between an ER and the CCS (or an MC) are shown in Figure 4.3 by messages 3 and 5 respectively. The input is the location vector for a particular W_i and the specific location for which ERs seek information. The output is a set of MI_u associated with W_i.

(a) For a W_i, an ER follows the step 1 of BuildIndex to calculate T_{W_i} and L_{W_i}, then sends (4.8) in an encrypted form to the CCS/MC.

$$Q\{L_{W_i}\} = \{y_1 = f(\gamma, z_1), ..., y_r = f(\gamma, z_r)\}. \tag{4.8}$$

(b) Then, the CCS/MC finds a set ν as shown in (4.9) and sends it back to the ER.

$$\nu = \bigcap_{i=1}^{r} B_{H_i(y_i)} \qquad \forall B_{H_i(y_i)} \in SBF. \tag{4.9}$$

(c) The ER will decrypt each element of ν to find MI_u for all the DOs with the same keyword. If necessary, the ERs are able to retrieve the complete PI using $CS_x||\sigma_u$.

4.3.3 Data Entry Updating

The proposed algorithm allows DOs to update their entries to the SBF. The updating process can be triggered when a DO would like to add or remove a keyword to/from the

index and its corresponding entries in the SBF. In addition, a DO is able to update the SBF if he/she moves to another location area.

The addition and removal processes are supported in our construction using the CBF_u and OBF. When adding a keyword is required, a DO runs the BuildIndex algorithm and sends the result to the CCS to be stored in SBF and updates her CBF_u. On the other hand, to remove a keyword, the DO calculates a removal BF (RBF_u) following the same procedures. Then, the DO compares RBF_u with CBF_u to see if any index in the RBF_u has a value more than one in the corresponding index in CBF_u. Here, the ones for which CBF_u has a value more than one, the user first flips the bit from 1 to 0 in RBF_u, then picks a random OE from OBF where its corresponding value in CBF_u is zero and updates RBF_u with that to obtain RBF'_u. Note that the DO should also update CBF_u using RBF_u by decrementing the corresponding r elements by 1. OBF should also be updated if necessary. Finally, the DO sends the removal request along with RBF'_u and $\xi'\{\xi(MI_u)||C(BF_u)\}$ to the server.

4.3.4 Conjunctive and Disjunctive Search

Our scheme fully supports conjunctive search queries while it faces certain limitations for disjunctive queries. For conjunctive queries of multiple keywords, an ER runs the *BuildIndex* algorithm on all the keywords and calculates a single BF_u. Instead of sending L_{W_i}, the ER sends $\xi'\{BF_u\}$ to the CCS/MCs using SSL. The *SearchIndex* finds the intersection of indexes in the SBF that are marked in the query and sends back the result. Figure 4.5 illustrates the conjunctive search.

In the case of the disjunctive search queries, we need to modify BF_u so that the server is able to distinguish distinct keywords. For this purpose, we employ a CBF. Here, every bit in the BF_u is substituted with a counter. In other words, every element of the index is represented with $C_{len} \geq 1$ bits where C_{len} is the length of a counter. Using the counter, we can assign a distinct value in the range of the counter to a distinct keyword. This way, the CCS/MC can perform disjunctive search queries. Figure 4.6 shows an example where the search query is comprised of three distinct keywords. In this example, $C_{len} = 2$ bits.

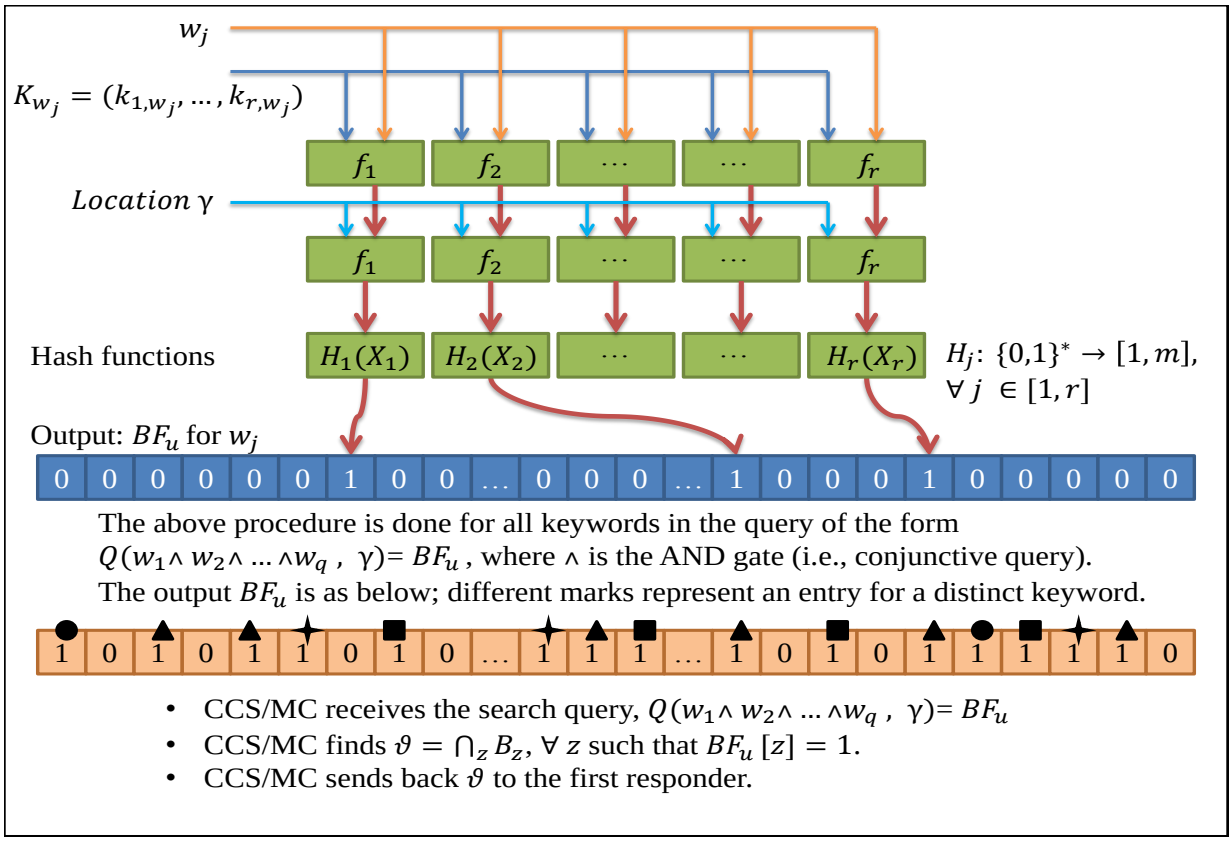

Figure 4.5: Conjunctive keyword search algorithm

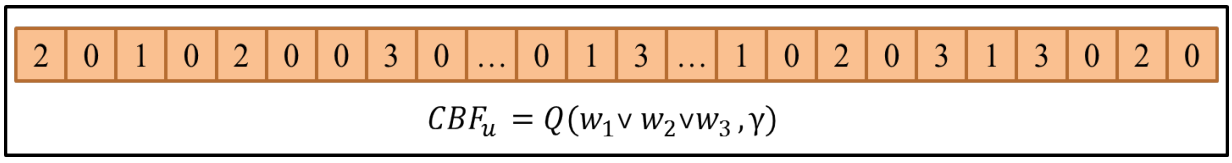

Figure 4.6: Disjunctive keyword search query

To apply the proposed disjunctive search algorithm, there is a limitation which is the fact that if the number of distinct keywords in the disjunctive query is large, the search query index cannot fully represent all of the keywords distinctively. This is because one or more mappings of distinct keywords in the index may overlap as shown in (4.10) and (4.11) with a certain probability. Note that $H_i(y_{j,i}) = H_{i'}(y_{j',i'})$ implies that the two distinct inputs point to the same buffer (i.e., an overlap occurs).

$$L_{W_j} = \{y_{j,1}, ..., y_{j,r}\} \qquad \forall j \in Q \tag{4.10}$$

$$\exists\{j \neq j'\} \in Q \rightarrow H_i(y_{j,i}) = H_{i'}(y_{j',i'}) \qquad \text{for } i, i' \in [1, r] \tag{4.11}$$

Considering the size of a CBF (i.e., $|CBF_u| = m$ counters), such an event (as represented in (4.10) and (4.11)) occurs with probability p_{overl}, after updating $n(p_{overl}, m)$ counters (i.e., to a value greater than 0) in the CBF_u. $n(p_{overl}, m)$ is approximated in (4.12) [137]. Setting $p_{overl} = 0.5$ (similar to the Birthday paradox), we can find an upper bound for the number of keywords N_W to include in the search query as shown in (4.13).

$$n(p_{overl}, m) \approx \sqrt{2m \cdot \ln\Big(\frac{1}{1 - p_{overl}}\Big)}. \tag{4.12}$$

$$N_W + 1 \leq \lceil \frac{n(0.5, m)}{r} \rceil \tag{4.13}$$

The upper bound in (4.13) limits the number keywords included in the disjunctive keyword search and bounds it such that with low probability an overlap would occur. However, beyond the threshold in (4.13), one can simply modify the BuildIndex algorithm such that the value of a counter in the CBF_u represents the last indexing operation. For example, if a counter is the target of two keywords, the counter only represents the second keyword. This implies that for at least one of the distinct keywords in the search query there might be $z < r$ counters that have the same value. In this case, the SearchIndex algorithm should also be modified such that the CCS/MC returns ν for a particular counter value (which corresponds to a particular keyword), even if the intersection is done for $1 \leq i \leq r$ as opposed to the one shown in (4.9).

The shortcoming of this approach is that the accuracy of the result decreases since the SearchIndex checks $i < r$ buffers. In this case, the false positive probability increases with the ratio shown in (4.14). In this equation, r' is the number of the counters left after overlapping for a specific entry.

$$fp_{ratio} = \frac{(1 - e^{-r'l/m})^{r'}}{(1 - e^{-rl/m})^{r}}. \tag{4.14}$$

4.4 Security Analysis

Our algorithm is semantically secure against a Chosen Keyword Attack (CKA): an attacker cannot learn anything about a set W_u from its BF_u for two main reasons. First, we use HMAC as our pseudo-random function for which an adversary has negligible advantage to break that. In other words, an adversary has negligible advantage to distinguish if f (i.e., HMAC) is a random function or a pseudorandom function. Moreover, we used different nonce and secret key for each HMAC function to differentiate various HMAC functions used in the algorithms. In addition, HMAC is a one-way function as mentioned in Chapter 3. Furthermore, comparing two BF_us, an attacker will not learn which index contains more W_j than the other. This is because both include the same number of elements which is the result of adding OEs to the index.

In the CKA game, the challenger performs the setup algorithm and makes available the set of keywords and location areas. The challenger also generates two sets of secret keys and nonces corresponding to each keyword and HMAC respectively. The challenger keeps these two sets secret. An attacker starts a query phase in which he/she sends several queries (up to some threshold) to the challenger and receives back corresponding indexes. Then, an attacker chooses two keywords and submits them to the challenger. Then, the challenger generates the index for the two and flips a coin and sends back the index corresponding to the coin flip result. Here, an attacker has negligible advantage to determine which keyword was chosen for which the index was generated. The security of HMAC has been proven in [138, 139, 140].

Considering our threat model in which a server is assumed to be honest but curious, we need to investigate the proposed algorithm in terms of privacy preservation. The privacy of the proposed algorithm has a direct relationship with the security of the algorithm mentioned above. Here, we need to investigate the probability of an event ϖ in which for

any two users a and b with distinct sets of keywords $W_{u_a} \neq W_{u_b}$, after indexing $2q$ elements into an SBF, at least r elements of user a intersect with r elements of user b in the SBF. In other words, at least r elements of user b are placed into the same r distinct buffers where r elements of user a were placed. From the server's point of view, this event implies that the two users might at least have one keyword in common. Higher $\Pr(\varpi)$ makes more confusion for the server, thus providing more privacy. This probability is presented in (4.15).

$$\Pr(\varpi) = 1 - \sum_{k=0}^{r-1} \frac{\binom{\lambda}{k} \times \binom{m-\lambda}{\lambda-k}}{\binom{m}{\lambda}}, \tag{4.15}$$

For example, if $l = 100$, $r = 10$, $|\gamma| = 1$, $q = 15$, then, $m = 1443$ and $\lambda = 142$ based on (4.2). Thus, $\Pr(\varpi) \simeq 91.5$ percent. This means that even if r elements of two distinct BF_us intersect, with probability of 91.5 percent, those belong to two distinct sets of keywords. In (4.15), the complement of the event (i.e., less than r elements overlap) is subtracted from 1 to obtain $\Pr(\varpi)$. The term $\binom{m}{\lambda}$ shows the number of possible ways to choose λ buffers in an SBF that has m buffers. In addition, the term $\binom{\lambda}{k} \times \binom{m-\lambda}{\lambda-k}$ shows that k buffers in the SBF are selected from λ buffers in the SBF containing the elements of user a to insert the elements of the user b, and the rest (i.e., $\lambda - k$) are selected from the empty buffers (i.e., $m - \lambda$).

In our scheme, a CCS or MC is not able to deduce which buffers in SBF are the target for a specific keyword. Unlike the work in [141] which suffers from a dictionary attack, our algorithm prevents a dictionary attack, as a result of using HMAC which is a keyed hash function. In this regard, only authorized DOs can generate legitimate indexes. In addition, DOs receive the keys corresponding to their keywords set. Therefore, the DOs cannot generate any index beyond such a set.

Last but not least, confidentiality of MI and search queries are provided via an encryption algorithm. In addition, compared to the work in [80], our algorithm does not rely on a private server in order to build an index, generate search queries, and provide privacy.

4.5 Performance Analysis

In this section, we will explore communication overhead, computation complexity, and memory usage. We simulated our scheme using the Java programming language on a desktop computer with the Ubuntu operating system. The PC is running on a Core i3 CPU with a processing speed of 3.3 GHz.

4.5.1 Communication overhead of the Scheme

The communication overhead from a DO to a CCS/MC is $|\xi'\{\xi(MI_u)||C(BF_u)\}|$. The maximum size of $|MI_u|$ is obtained when $|W_u| = q$. For every element in W_u and the set $\{U_{P_{ID}}, CS_x, \sigma_u, PS_a\}$, we use 160 bits representation (e.g., using SHA1 as the generator). Also, we use ECC 256 bits for $\xi()$ and AES 128 bits for $\xi'()$. Suppose $q = 15$, $|BF_u| = 30$ Kbits, and $r = 10$, then $C(BF_u)$ (the compression function that was explained before) can decrease $|BF_u|$ by approximately 92 percent. Therefore, the maximum communication overhead will be less than 6 Kbits. Note that the low communication overhead from the user to the server along with addition and removal capabilities of our system, enable our scheme to be compatible with dynamic scenarios where individuals constantly move from one location to another and need to update the SBF.

The communication cost from an ER to a CCS/MC depends on the construction of queries and the number of keywords included in a query (i.e., N_W). In this case, two cases should be taken into consideration: without the multi-keyword search, and with the multi-keyword search. For the former, each keyword is searched separately for which the results are received. In this case, the communication overhead is proportional to the number of keywords per query and the size of a query (i.e., $N_W \times |C(BF_{W_i})|$). Suppose the compression function is defined as follows: instead of sending the entire BF bit string, one sends the bit indexes where the bit flipped from 0 to 1. Therefore, the communication overhead will be $N_W \times r \times \log_2 m$ bits where $\log_2 m$ is the required number of bits to represent a bit index.

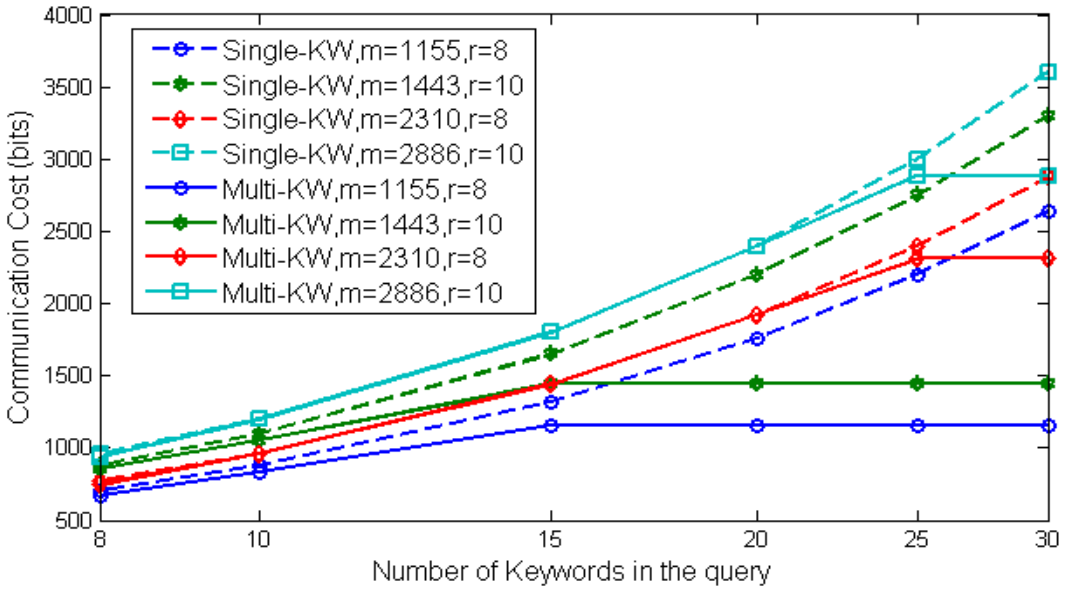

Figure 4.7: Query Communication cost (Single-keyword vs Conjunctive keyword)

Employing the multi-keyword search with conjunctive queries, we only send one request to the CCS/MC for the entire query that is comprised of several keywords. Therefore, without the compression function, the communication cost from an ER to the CCS/MC is constant (and equals to $|BF_u|$). On the other hand, using the compression function, the communication cost depends on the number of keywords in the query. Here, the maximum communication cost is imposed when the number of keywords in the query passes the threshold in (4.12). After this point (which is $N_W \geq \frac{m}{r \times \log_2 m}$), the compression function does not decrease the communication cost, thus, the communication overhead reaches the same value as if no compression function was used (i.e., $|BF_u|$).

Figure 4.7 illustrates the aforementioned situation using simulation results. Here, the results are within the 95% confidence interval. As this figure indicates, multi-keyword conjunctive search is most effective when the BF is more compact (i.e., m is small). Besides, for $m = 1155$ and $r = 8$, if the conjunctive query is comprised of more than 10 keywords, the multi-keyword query is more efficient than the single-keyword query. For larger m, both queries impose the same communication cost until the $r \times NW \geq n(p, m)$ point from which the multi-keyword search query is more efficient than single keyword search.

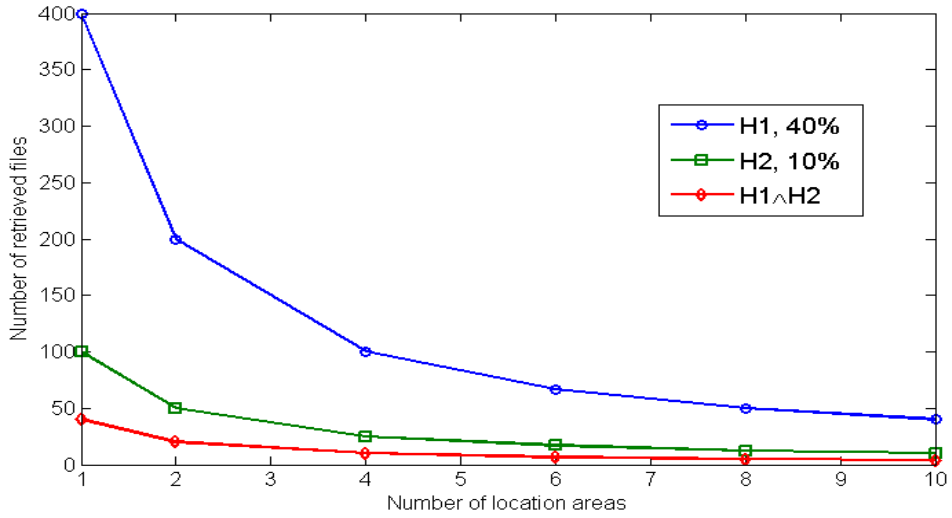

Figure 4.8: The effects of the number of locations and conjunctive keywords on communication cost from the CCS/MC to an ER

The main advantage of conjunctive keyword search shows itself in the communication overhead from a CCS/MC to an ER. For a single-keyword query, the communication overhead from a CCS/MC to an ER is proportional to the amount of data tagged with the queried keyword in a specific location area (i.e., $|t_{reg,W_i}|$ where t_{reg,W_i} is the number of registered users who have used the same keyword and location area to build their index). Thus, the overhead equals $|t_{reg,W_i}| \times |\xi'\{\xi(MI_u)\}|$. However, using the conjunctive keyword search, the amount of retrieved data may decrease. This is because the conjunctive search query results in the common elements of two or more distinct BF_us. For example, if two DOs have outsourced their data for two different sets of keywords (e.g., W_{Alice} =$\{Asthma, A^+\}$ while W_{Bob} =$\{Handicap\}$), the conjunctive keyword search on $Handicap \wedge Asthma$ returns nothing.

Figure 4.8 illustrates the effect of conjunctive keyword search on the amount of retrieved data (i.e., communication overhead from a CCS/MC to an ER). Suppose in a system $1 \leq |\gamma| \leq 10$ and there are 1000 DOs that are distributed uniformly in those location areas. Suppose health attributes H_1 and H_2 exist among 10 and 40 percent of people respectively. It is also assumed that H_1 and H_2 are independent from one another. To retrieve the data for $H_1 \wedge H_2$, considering a single keyword query, an ER sends two separate queries for

each attribute in the example. Here, the result includes the combined amount of data tagged with either one of the keywords (in this example, the number is expected to be approximately 500 data files for one location area). Then, the intersection of the received data will be the result which is done at the ER side. On the other hand, using conjunctive keyword search for the same example, the amount of retrieved data will be decreased by approximately 92 percent. This shows the effectiveness of our conjunctive keyword search scheme.

It is worth mentioning that γ has a significant influence on the amount of retrieved data (i.e, $|t_{reg,W_i}|$). Suppose that there are $|\gamma| > 1$ specified locations. Thus, two DOs with the same health attribute (e.g. Asthma), but in different location areas, would target different buffers in the SBF with high probability. Thus, searching for Asthma in one of the location areas only retrieves the corresponding DO. This results in more accurate data retrieval. Moreover, the communication overhead is decreased as well. Figure 4.8 also shows the impact of the location parameter on the communication cost. The figure indicates that increasing $|\gamma|$ from 1 to 10 decreases the communication overhead by 90 percent for different queries.

Employing the disjunctive keyword search, the communication cost from an ER to the CCS/MC is decreased. This effect is without the consideration of a compression function. Here, for a number of distinct keywords q_{dj}, the single keyword search imposes $q_{dj} \times |BF_u|$-bit communication cost while a disjunctive keyword query requires $\log_2^{q_{dj}} \times |BF_u|$ bits. In the reverse direction (i.e., from the CCS/MC to the ER), comparing the single keyword and disjunctive keyword queries, the latter may require lower communication cost since the intersections among the data can be sent only once.

There is another important aspect that should be taken into consideration which is the fact that a high number of OEs may affect the accuracy of the results and increase the communication overhead from the CCS/MC to an ER. Here, we investigate the probability of an event (we call it overlapping probability) in which if a DO uses a number of OEs to obfuscate his/her index, a fraction of the total number of OEs fully collides with at least one of the l keywords. In other words, what would be the probability that any r number

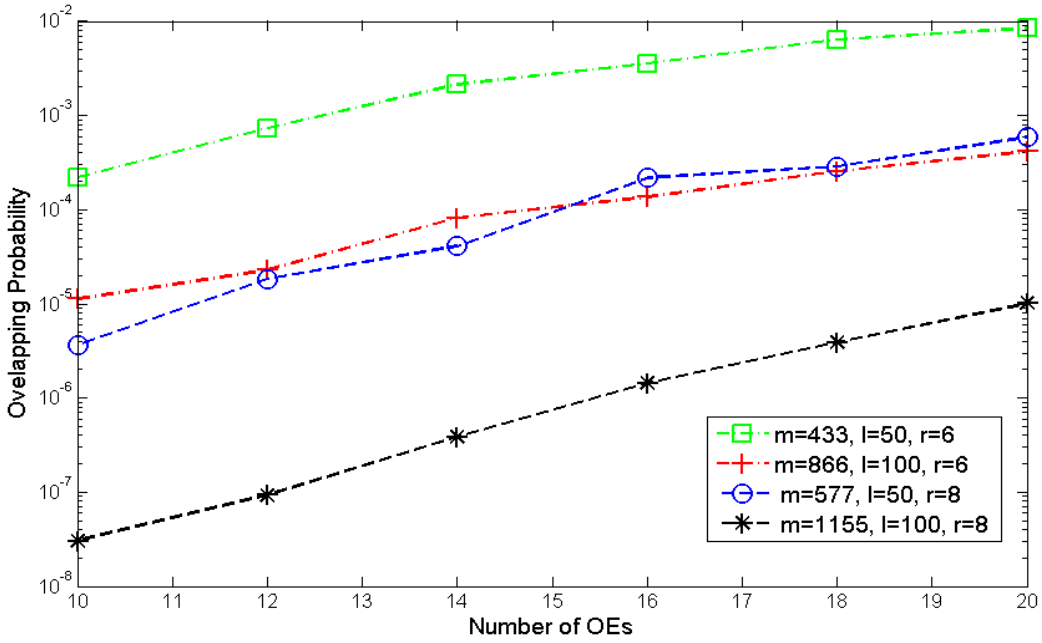

Figure 4.9: Overlapping Probability

of similar OEs in an SBF intersect exactly with the r elements of at least one indexed keyword? We performed simulation analysis to project this probability as shown in Figure 4.9. For each point in the simulations, the number of simulations was chosen so that the 95% confidence interval is negligible.

In Figure 4.9, r and l take two different values. The figure illustrates that when the number of OEs increases, then overlapping probability increases. The maximum probability of approximately 0.8 percent occurs when $l = 50$, $m = |SBF| = 432$, $r = 6$, and we added $|OE| = 20$ to SBF. When r is constant but l and $|SBF|$ increase, the overlapping probability decreases. In addition, when r increases, for the same l, the probability decreases. For the number of OEs less than 10, the overlapping probability is less than 10^{-5} for $m > 577$. This result shows that the use of OEs in our scheme does not interfere with accuracy of search results and provides good levels of privacy.

Notice that for $|\gamma| > 1$, to keep the false positive rate unchanged, we need to modify (4.2) as in (4.16). In this case, the overlapping probability decreases since m increases (as shown in Figure 4.9). Here, having lower overlaps does not imply that two DOs have distinct sets of keywords. This is because even two identical sets of keywords that were indexed for two distinct location areas would have distinct indexes with high probability. Therefore, considering our threat model for a server, the privacy is still preserved against this entity.

$$m = \frac{lr|\gamma|}{\ln 2} \tag{4.16}$$

4.5.2 Memory Requirements of the Scheme

The memory usage at CCS/MC equals $M = |SBF| \times \beta \times \tau$, where $|SBF| = m$ is the length of an SBF and β is the maximum amount of data inserted in one buffer, and $\tau = |\xi'(MI_u)|$ is the maximum size of each data file in a buffer. Before we quantify M, we need to measure the probability of an event in which one buffer overflows. This is done using simulation analysis. The results are within the 95% confidence interval.

Figure 4.10 shows the buffer overflow probability. The figure depicts the buffer size requirements when the number of DOs increases from 500 to 1000 individuals and $|\gamma|$ increases from 5 to 20. Suppose $l = 100$, $r = 10$, and $|\gamma| = \{20, 5\}$, thus $|SBF| = \{28854, 7214\}$ respectively. It is immediate that if buffer size increases, the probability of overflow decreases. Figure 4.10 *a* shows the change in the buffer overflow probability when $|\gamma| = 20$. Here, when $t_{reg} = 500$ and $\beta = 20$, the overflow probability is approximately 67 percent. However, when β increases to 35, the overflow probability drops to approximately 10^{-4}. For $t_{reg} = 1000$, the overflow probability is 1 until $\beta = 35$. But, increasing β to 50 makes the overflow probability drop down to approximately 2.7×10^{-4}.

Figure 4.10 *b* depicts the situation where $|\gamma| = 5$. For $t_{reg} = 1000$, using $\beta \geq 280$ the overflow probability decreases from 1 to approximately 10^{-4} for $\beta = 320$. Comparing the two graphs in Figure 4.10 shows that when $|\gamma|$ raises 4 times, the buffer size requirements

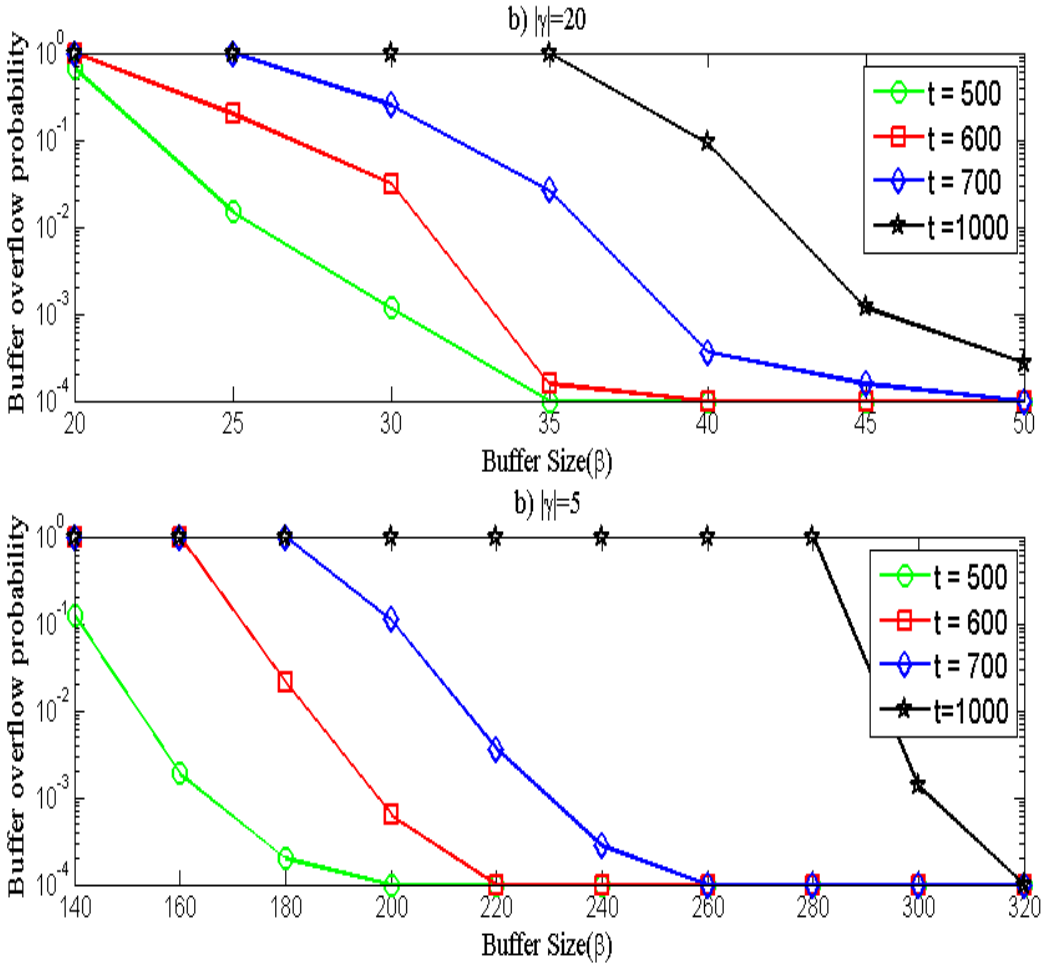

Figure 4.10: Probability of buffer overflow, a) $|\gamma| = 20$, b) $|\gamma| = 5$

fall approximately more than 5 times. Therefore, suppose $t_{reg} = 600$, $\tau = 5Kbits$, $|\gamma| = 20$, $l = 100$, $r = 10$, and $\beta = 50$, $M \simeq 881MB$. This result shows that our secure data storage structure is affordable even to be deployed in MCs.

4.5.3 Computational Overhead and Delay

Table 4.1 shows the computational complexity for each procedure of our proposed scheme in comparison with the work in [80] which is most relevant to ours. The procedures are at both the user side and server side. In terms of computational complexity, we used HMAC in building the index and search procedure which was done in a distributed way with very

Table 4.1: Computation Complexity

Scheme	Buildindex (t_{reg} DOs)	Search (ERs)	Search (CS)	Add (DO/CS)	Remove (DO/CS)
Ours	$t_{reg} \times [(q \times 3r)H + \xi]$	$rH + \lvert t_{reg,W_i} \rvert \times D_\xi$	$rH + I$	rH	rH
[80]	$t_{reg} \times [(2q+2)PRP + q \times \xi]$	$2PRP$	$I + \lvert t_{reg,W_i} \rvert \times D_\xi$	$2PRP + \lvert t_{reg,W_i} \rvert \times D_\xi + \lvert t_{reg,W_i} + 1 \rvert \times \xi$	$2PRP + \lvert t_{reg,W_i} \rvert \times D_\xi + \lvert t_{reg,W_i} - 1 \rvert \times \xi$

low delay. However, the authors in [80] use a private server to which the computations of t_{reg} registered individuals are outsourced. H is used to show HMAC computation and PRP stands for a pseudorandom permutation function such as AES. ξ, D_ξ are used to show encryption and decryption processes respectively. Finally, I is used to show the intersection operation between r buffers.

According to Figure 4.10, increasing γ decreases β requirements, which implies that the number of intersection operations and search results will also be decreased. Consequently, the amount of computations decreases. Furthermore, our scheme imposes very low computation burden for addition and deletion processes in comparison with the work in [80] in which the public server needs to decrypt an entire linked list and then modify it for any single alteration. This indicates the applicability of our scheme for dynamic situations where the cost of updating needs to be limited.

In general, the data retrieval process consists of two procedures: search over encrypted data, and decryption. In our scheme and [80], search takes place merely over the number of data files containing the keyword (i.e., $O(1)$ delay) and not the entire database (i.e., $O(n)$ delay). Note that $O(1)$ delay has a significant impact on the total data access delay under critical circumstances where the size of a database is large or an immediate response is required. For further comparisons on search delay refer to Table 2.2 in Subsection 2.5.3. Our scheme decrypts ECC and AES ciphertext messages, whereas the methods in [78, 68, 94, 95, 96, 97, 98, 99, 100] involve Pairing Based Cryptography (PBC) which requires more computational resources.

4.6 Conclusion

In emergency situations, privacy preservation, context, and location-aware information is required. Existing works did not address such requirements in PSBNs. In this chapter, we proposed a storage bloom filter and modified a secure index algorithm to provide data availability with regards to PSBN requirements. Our search process imposes $O(1)$ delay which is ideal for emergency situations. In addition, communication complexity is very low from a DO to a CS and it is proportional to the number of data files containing the search query in the reverse direction. The memory usage is also affordable even for MCs with limited resources. We used a location parameter γ with which we decreased the buffer size and the number of search outcomes. The latter decreases communication and computational complexities and delay.

Chapter 5

Location-Aware Authorization Scheme

5.1 Introduction

Effective emergency (such as a hurricane, building on fire, etc.) response requires accurate, relevant, timely, and location-aware information (e.g., environmental information, health records, etc.). Acquiring information in such critical situations encounters substantial challenges such as a large volume of data processing, unstructured data, privacy, and authorized data access. In Chapter 4, we proposed a privacy-preserving search algorithm to address data availability. The proposed algorithm provides accurate, timely, and location-aware information for ERs. However, data access authorization was limited in the sense that it was not flexible and fine grained. In addition, the authorization was delegated completely to a trusted third party.

As discussed in Chapter 2, existing solutions for data access authorization either do not scale well or merely consider a Break-the-Glass method in which a master key is provided to ERs to decrypt a ciphertext [142]. For example, employing ABE a DO may merely use an *"emergency"* attribute in the access policy and generate the corresponding ciphertext. This solution may enable unauthorized users to access data (i.e., the ones who did not have access to the data before an emergency). In fact, access to data (especially PI) should only

be authorized if its owner is somehow involved in the emergency incident. In addition, the Break-the-Glass method may cause ERs to become overwhelmed by the large volume of accessible data. In other words, this method is not capable of filtering irrelevant data. Therefore, the Break-the-Glass method is impotent to respect such requirements.

To jointly address the aforementioned issues, in this chapter we propose a location-aware authorization scheme which protects privacy and provides flexible and fine-grained access authorization. Moreover, the proposed scheme filters irrelevant data by taking into consideration the time and location of the ongoing emergency. This requires incorporating dynamic attributes (i.e., location and time) into the authorization scheme. Since location and time are dynamic, whenever they change, the ciphertext should also be updated.

To construct such a scheme, we propose to employ CP-ABE. Using CP-ABE, a DO is able to enforce his/her preferred access policy into ciphertext. However, movements of a DO to different locations in addition to the changes of time may result in a large number of ciphertext updating messages. To tackle such an issue, we innovatively incorporate CP-ABE with Broadcast Encryption (BE) and construct our novel scheme called Location-Aware Ciphertext-Policy Attribute-based Encryption (LA-CP-ABE).

We use BE in an unconventional way to incorporate the location attribute into an access policy. In this case, we broadcast a message to a set of locations instead of individuals. On the other hand, we delegate the access authorization based on the time attribute to a cloud server. We assume the cloud server is on-line all the time. Therefore, when an emergency happens, the Public Safety Answering Point (PSAP) sends an emergency trigger message including the time of emergency occurrence, the time interval in which ERs' queries are considered valid, and the location area of the incident. When a cloud server receives a query, it checks the validity of the query generation time, and if it was within the allocated time interval, the server updates ciphertext accordingly and sends it to the ER. The new ciphertext is only valid for a specified time interval.

Employing our proposed scheme, a DO does not need to delegate the entire authorization process to a trusted third party server. Furthermore, integrating BE with CP-ABE to enforce the location attribute, and delegating the control on the time of access to an

on-line server, decreases the frequency of ciphertext updating messages. As a result, our proposed scheme provides authorized access to accurate, relevant, timely, and location-aware information. We provide extensive security analysis and performance evaluations to demonstrate the effectiveness of our scheme. The analysis shows that the scheme imposes constant communication and decryption computation overheads. Furthermore, the proposed scheme is proven Chosen Ciphertext Attack (CCA) selectively secure based on the $m-$Bilinear Diffie-Hellman Exponent ($m-$BDHE) assumption. It also addresses the key escrow problem.

The remaining sections are as follows. Section 5.2 introduces the preliminaries of our work. The system model, threat model, and assumptions are presented in Section 5.3. Challenges of designing LA-CP-ABE are discussed in Section 5.4. Section 5.5 presents the LA-CP-ABE scheme. Security analysis and performance evaluation are discussed in Sections 5.6 and 5.7 respectively. Section 5.8 summarizes the chapter.

5.2 Preliminaries

In this section, we provide the details about the underlying tools and algorithms that are used in our system. In addition, the intractability assumption of our LA-CP-ABE scheme is explained.

5.2.1 Composite-Order Bilinear Groups

We construct our LA-CP-ABE scheme using composite-order bilinear groups [107]. A group generator function $\mathcal{G}$ takes as input the security parameter S_{sec} and outputs a description of a bilinear group $\mathbb{G}$. We define $\mathcal{G}$'s output as $(N, \mathbb{G}, \mathbb{G}_T, e)$, where $N = p_1 p_2$ is a product of two distinct primes (p_1 and p_2), $\mathbb{G}$ and $\mathbb{G}_T$ are cyclic groups of order N, and $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ is a map that is

1. Bilinear: $\forall g, h \in \mathbb{G},\ a, b \in \mathbb{Z}_N, e(g^a, h^b) = e(g, h)^{ab}$
2. Non-degenerate: $\exists g \in \mathbb{G}$ such that $e(g, g)$ has order N in $\mathbb{G}_T$.

We assume that the group operations in both $\mathbb{G}$ and $\mathbb{G}_T$ and the bilinear pairing map e are computable in polynomial time with respect to S_{sec}. Suppose that G_{p_1} and G_{p_2} are the subgroups of order p_1 and p_2, respectively. In a composite-order bilinear group, there exists an orthogonality property as follows: if $h \in G_{p_1}$ and $h' \in G_{p_2}$, $e(h, h') = 1_T$ where 1_T is the identity element in $\mathbb{G}_T$. To show this, suppose g is a generator of group $\mathbb{G}$. Then, g^{p_1} generates G_{p_2} and g^{p_2} generates G_{p_1}. Therefore, suppose for some x, y, $h = (g^x)^{p_2}$ and $h' = (g^y)^{p_1}$. Then,

$$e(h, h') = e(g^{xp_2}, g^{yp_1}) = e(g^x, g^y)^{p_1 p_2} = 1 \tag{5.1}$$

5.2.2 Anonymous Key Agreement

An anonymous one-way key agreement is proposed in [143] using bilinear maps. The algorithm guarantees sender-side anonymity as a result of non-interactive key agreement. Considering our application, this is an important feature because preserving privacy of an ER's actions (i.e., data requests) from the cloud server requires that the linkage between the identity of the ER and his/her actions is broken [144]. This linkage can be broken by hiding the identity of the ER. Sender-side anonymity also can protect DOs' privacy. This is because the identity/role of an ER (e.g., Bob/Policeman) may reveal some information about a DO.

In most one-way anonymous communications, authenticating a non-anonymous server is needed. Here, using this algorithm, the shared key is implicitly authenticated. In other words, the sender is assured that only the server can compute the key. Suppose there is a cryptographic hash function $H : \{0,1\}^* \rightarrow \mathbb{G}$. An authority generates a master secret key s, uses a public identity of a recipient $ID_{\mathbb{B}}$ along with the sender's private key $SK_{\mathbb{A}} = Q_{\mathbb{A}}^s = H(ID_{\mathbb{A}})^s \in \mathbb{G}$, and generates a session key as follows:

1. Sender $\mathbb{A}$ computes $Q_{\mathbb{B}} = H(ID_{\mathbb{B}}) \in \mathbb{G}$. $\mathbb{A}$ chooses a random number $\alpha \in \mathbb{Z}_P$ where P is the order of $\mathbb{G}$, and generates the pseudonym $P_{\mathbb{A}} = Q_{\mathbb{A}}^{\alpha}$ and sends it to the receiver $\mathbb{B}$. Then, $\mathbb{A}$ generates the session key $k = e(Q_{\mathbb{B}}, SK_{\mathbb{A}})^{\alpha} = e(Q_{\mathbb{B}}, Q_{\mathbb{A}})^{s\alpha}$.

2. Recipient $\mathbb{B}$ computes the session key using $SK_{\mathbb{B}} = H(ID_{\mathbb{B}})^s$ as follows,

   $k = e(P_{\mathbb{A}}, SK_{\mathbb{B}}) = e(Q_{\mathbb{A}}, Q_{\mathbb{B}})^{s\alpha}$.

5.2.3 Complexity Assumption

The complexity assumptions for our system are based on the decisional Bilinear Diffie-Hellman Exponent assumption (BDHE). Recall that $\mathbb{G}$ is a bilinear group of composite-order N. The $m-$BDHE problem in $\mathbb{G}$ takes $(h, g, g^a, g^{(a^2)}, ..., g^{(a^m)}, g^{(a^{m+2})}, ..., g^{(a^{2m})}) \in \mathbb{G}^{2m+1}$, as input and outputs $e(g, h)^{(a^{m+1})} \in \mathbb{G}_T$. Suppose, $g_i = g^{(a^i)} \in \mathbb{G}$. We say an algorithm $\mathcal{A}$ has advantage ϵ in solving $m-$BDHE in $\mathbb{G}$ if $\Pr[\mathcal{A}(h, g, g_1, ..., g_m, g_{m+2}, ..., g_{2m}) = e(g_{m+1}, h)] \geq \epsilon$, where the probability is over the random choice of generator g in $\mathbb{G}$, the random choice of h in $\mathbb{G}$, the random choice of a in $\mathbb{Z}_N$, and the random bits used by $\mathcal{A}$.

5.3 System model and Threat model

It is assumed that a city is divided into n distinct location areas with equal areas, each having a unique pseudo-identity L_{id}. We choose a cloud server storage model to maintain data and perform the data updating procedure. The system is comprised of several entities as follows.

KGA: This entity generates the secret keys of the system and performs the setup algorithm. It is assumed that there are two separate KGAs; one is for location and time attributes, and the other is for the rest of the attributes, introduced in Section 5.5.

PSAP: This entity receives an emergency signal including 9-1-1 calls and sensor signals (e.g., smoke detectors, heat detectors and so forth), and triggers the cloud servers and ERs accordingly.

CS: We assume that there is a central cloud server which stores all encrypted PI.

DO: This entity is a member of the general public who registers to the system by communicating with KGAs and uploads his/her encrypted PI to the CS.

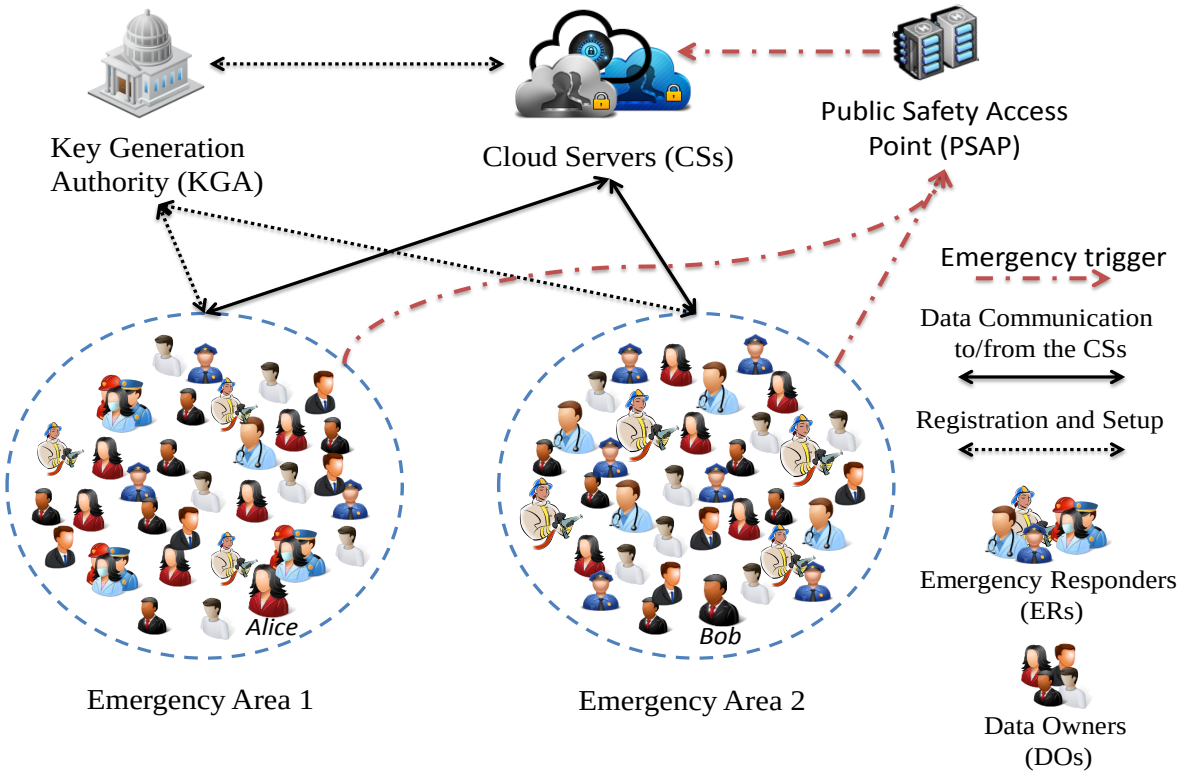

Figure 5.1: System model

ERs: These are the governmental authorities including policemen, fire fighters, and paramedics. They also will register to the system by communicating with the KGAs and receive the system parameters and their secret keys.

As depicted in Fig 5.1, there is an area (e.g., Emergency Area 2) in which an emergency incident has occurred. Note that before an incident happens and during normal conditions, those DOs who have registered to the system outsource their encrypted PI to the CS. The goal of this work is to provide authorized access to location-aware data for ERs. It is assumed that an agent is equipped with a smartphone which has a tamper-proof GPS. Such a tool has secure components to perform simple calculations and secure storage [145, 146]. The user cannot access the secure component of GPS, and it is assumed that GPS performs honestly. The communication between the users in the system and KGAs/CS can be facilitated using WiFi, 2G, 3G, etc.

We assume that KGAs are fully trusted, but the CS is honest but curious. This means that the CS follows the procedure of the scheme in an honest way, but tries to learn as much information as possible. We assume that KGAs authenticate DOs and ERs and only then it transfers secret keys to those entities. The authentication procedure is out of the scope of this work, but it can also be provided using well-known methods [147, 148, 149]. We also assume that there exist eavesdropper adversaries who live among the general public and would like to learn as much information as possible.

In this work, a data access model is also proposed. Since any emergency is related to a location and occurs at a certain time, our model enables authorized access to victims' information at the time of an emergency from a predefined distance to the emergency scene. This model ensures the data access is authorized and a DO is involved in an emergency, and at the same time filters irrelevant information that is available to ERs. Consider Fig 5.1: the information of Bob who is located in Emergency Area 2 may not be useful for an ER who is located inside Emergency Area 1. This way the level of data accuracy and relevance to an emergency increases.

Finally, considering our access model as a threat model, if a user is in location area L_i at time τ_t, she/he is not able to access a DO's data if the DO is located in $L_{i'}$ for $i \neq i'$. In addition, the generated key for the L_i and τ_t is invalid for the same location at time $\tau_{t+t'}$ at which the DO is not located anymore (e.g., when there is no longer an emergency). This provides a higher level of privacy protection than the Break-the-Glass approach. We will further elaborate our model for incorporating location areas into our scheme in Section 5.5 where we will demonstrate that the proposed access model is flexible and does not prevent authorized users from accessing information when required.

5.4 Challenges in Designing Location-aware CP-ABE Schemes

Effective emergency response requires that communication overhead, and computation complexity/delay of the authorization scheme, be sufficiently small that authorized data

access is facilitated. Therefore, constant ciphertext-size CP-ABE schemes with constant computation complexity are suitable choices for authorization. Here, we sacrifice the flexibility and expressiveness of an access policy for the sake of better performance. However, incorporating dynamic attributes (i.e., location and time) into an access policy of this particular kind of CP-ABE is challenging. In this section, we will elaborate the corresponding challenges with regards to both DOs (as ciphertext generators) and ERs (as data consumers).

From a DO's perspective, there are three natural ways to include dynamic attributes into ciphertext. First, a DO could trivially predict the time/location and include them into ciphertext in the first place. This may sound easy as there are limited locations that a DO may visit per day. However, the DO may become anxious since she/he has to follow the predicted schedule. Furthermore, only one out of many choices of location and time attributes would be legitimate at any instant. In this case, the proper access policy category that would fit the preceding approach is $(t_{sh}, n)-$threshold. However, it is a complex task to combine both dynamic and static attributes into an access policy. This is because out of all attributes, 2 have to be specified for location and time and the rest are other attributes. Therefore, an ER possessing t_{sh} matching attributes excluding location and time may still be able to decrypt the message.

The second way of including dynamic attributes into an access policy is that a DO updates the ciphertext every time that she/he moves to another location or after the expiration of a time interval. The cost of such an approach grows linearly with the number of visited locations per day and number of time intervals set by the system. Besides, a DO may visit some similar locations several times a day in different time intervals which makes the updating process very inefficient. (Note that a DO can delegate such a process to a smartphone in order to automate the process.) Another drawback is the fact that in an emergency, a DO may be unconscious or the smartphone may be broken or lost which may render the updating process incomplete. Therefore, utter reliance on a DO has its own risks.

The third way is to delegate the entire updating process to the CS. This can be done in two ways: one, the CS decrypts the data and re-encrypts it using updated attributes; two, the CS privately updates the data using privacy-preserving proxy re-encryption techniques [150, 151, 152] in which decryption is not necessary. The first way poses a breach of privacy since the CS can access plain data. For the second way, although the CS may not be able to access plain data, DOs may still not want to trust the CS entirely with such a process. In general, total delegation of the authorization process requires ultimate trust in a server.

The aforementioned methods are either infeasible, costly, inefficient, or require ultimate trust in a third party. We propose a feasible and efficient way to incorporate dynamic attributes into an access policy. We delegate time of access authorization to the CS and the authorization for location and static attributes is enforced by DOs. To incorporate time of access, the CS checks the validity of a data request and then updates the ciphertext accordingly or rejects the request. Location authorization will be done by DOs. A DO will choose a set of preferred locations (that can be the most frequent locations in, e.g., a week such as *Home, School, Work place, Grocery store, etc.*) for which he generates ciphertext. Note that a DO does not need to predict the time at which he/she visits location areas in the preferred set. Using this technique relieves the DO from an unnecessary updating process every time he visits some common location which results in a decrease in the computation and communication overheads. It is worth mentioning that a DO still can update the ciphertext very efficiently if he moves out of all the locations in the preferred set. To implement this, we integrate BE with CP-ABE. In our scheme, BE is used unconventionally for locations instead of individuals. Each location has a unique ID. A sender broadcasts his/her data to n locations and an ER in one of those locations can decrypt data using his/her private key. Therefore, BE results in a $(1, n)-$threshold access structure.

From an ER's perspective, the challenge is to provide the ERs with proper secret keys corresponding to the dynamic attributes. Recall that BE schemes are preferred to be stateless meaning that the private keys should remain unchanged. However, updating ciphertext with new locations and time requires freshly generated private keys corresponding to those attributes. This introduces a contradiction between static BE private keys and

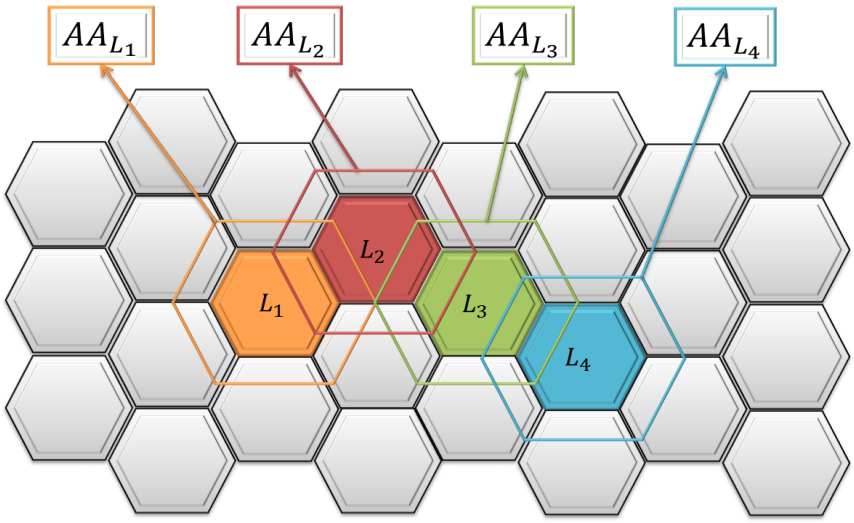

Figure 5.2: Location Area Model

the dynamic feature of location-awareness. There are two main options to overcome such a challenge. First, an ER sends a key updating request to a KGA specified for those dynamic attributes every time that she/he wants to decrypt data. Here, the KGA may become a single point of failure. Second, an ER can securely generate proper secret keys. This requires a tamper-proof device [153, 154].

5.5 Location-aware Ciphertext-Policy Attribute-based Encryption

Let us dive into the implementation of the LA-CP-ABE scheme. Here, we elaborate the location area model first, and then the construction of the scheme is presented. In the proposed scheme, DOs choose a set of location areas called *preferred location set*, S'', from the set of all possible location areas $\mathbb{L}$. S'' consists of a collection of location areas that a DO frequently visits such as *Home, School, Work place, Grocery store, etc.*

Since it is desirable for ERs to be able to access data before they arrive at the location scene (area) of an emergency, for each location area L_a, we define an *Associated Area* AA_{L_a} as depicted in Fig 5.2. Note that the hexagons used in Fig 5.2 are just for illustration purposes and do not mean that we assume a cellular network communication infrastructure.

The task of updating the ciphertext with new location areas is delegated to the DO's smartphone to automate the process. For instance, when a DO is in L_1, the AA_{L_1} has been incorporated to the ciphertext. In other words, a DO updates his/her ciphertext based on the boundaries of L_1, but an ER can have access to his/her data based on the wider boundaries of AA_{L_1}.

The diameter of an associated area could be chosen in such a way that the distance to the target location area gives the ERs sufficient time to retrieve the information before arriving at the scene. Here, there is a trade-off between privacy preservation, access authorization and data availability which depends on the drive time from the boundaries of a associated area to the emergency scene, geographical terrain of the area (e.g., urban area or rural area), data communication availability/reliability in that area, and so forth. Optimizing this diameter value is an interesting problem; however, it is out of the scope of this work.

Fig 5.2 also illustrates the trajectory of a DO from L_1 to L_2, then to L_3, and finally to L_4. For each change of a location, a DO needs only to remove/add one location area from/to the ciphertext. The proposed scheme does not need to regenerate the entire ciphertext when a DO changes his/her location. The scheme merely updates the original ciphertext by multiplying it with one group element by either adding or removing an attribute (refer to subsection 5.5.2 for more details). This is done with minimum communication overhead and low computation complexity. However, if the change of a location is among the ones in the S'', there is no need for ciphertext updating.

5.5.1 Construction of the Scheme

Let $\mathcal{G}$ be an algorithm that, on input security parameter S_{sec}, generates two groups of composite-order $N = p_1p_2$ with bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$. Let $H_K : \{0,1\}^* \to \mathbb{Z}_N$ be a keyed hash function and $H_1 : \{0,1\}^* \to \mathbb{G}$ a cryptographic hash function. Also, let $\mathbb{L} = \{L_1, L_2, ..., L_n\}$ be a set of all location areas; $U = \{A_0, A_1, A_2, ..., A_l\}$ be a universe of attributes where $|U| = l + 1$; $V_i = \{v_{i,1}, v_{i,2}, ..., v_{i,a_i}\}$ be a set of all possible values for attribute $i \in [1, l]$ and $a_i = |V_i|$; $W_{ER_u} = \{A_0, W_1, W_2, ..., W_\varrho\}$ is the attribute list of the ER_u where $W_i \in V_i$ and $\varrho + 1 = |W_{ER_u}| \leq |U|$. We assume A_0 is a default attribute shared

among all users of the system. We assume that KGA_1 generates the parameters for the static attributes, and KGA_2 generates the parameters for the location attribute. In this case, $\mathcal{G}(S_{sec})$ is run jointly by the two authorities. This means that KGA_1 and KGA_2 get the same description of $(\mathbb{G}, \mathbb{G}_T, e)$.

Setup$(S_{sec}, \mathbb{L}, (U, V))$**:** This algorithm is done by KGA_1 and KGA_2. In the following, $\xleftarrow{R}$ and $\in_R$ mean elements are assigned/chosen randomly from a group. For KGA_1, it takes as input S_{sec} and (U, V). It chooses a generator $g_1 \in G_{p_1}$, $q, R, \Lambda \xleftarrow{R} G_{p_1}$, and $\alpha, t_{i,j} \in_R \mathbb{Z}_N$ with $i \in [1, l], j \in [1, a_i]$. Note that Λ corresponds to A_0. KGA_1 computes $Y_0 = e(g_1, q)^{\alpha}$, $Y_2 = e(R, R)$, and $T_{i,j} = g_1^{t_{i,j}}$ for $\forall i, j$.

For KGA_2, it takes as input the security parameter S_{sec}, and the location set $\mathbb{L}$. It sets $\alpha', \beta, x \in_R \mathbb{Z}_N$, chooses a generator $g_2 \in G_{p_2}$, and $h_0, h_1, h_2, ..., h_n \xleftarrow{R} G_{p_2}^{n+1}$, where h_0 is a default parameter shared among all entities in the system, and h_i for $i \in [1, n]$ represents the AA_{L_i}. KGA_2 computes $Y_1 = e(g_2, g_2)^{\alpha'}$.

Finally, the public parameters PK (5.2) include a description of $(\mathbb{G}, \mathbb{G}_T, e)$ as well as

$$PK \leftarrow \left\{ g_1, q, R, \Lambda, Y_0, Y_2, \{T_{i,j}\}_{i \in [1,l], j \in [1,a_i]}, g_2, Y_1, h_0, h_1, ..., h_n \right\} \tag{5.2}$$

and $MSK = \{q^{\alpha}, g_2^{\alpha'}, \beta, x, \{t_{i,j}\}_{i \in [1,l], j \in [1,a_i]}\}$ is the set of private parameters where x is the master secret key allocated to the CS. Setup outputs (PK, MSK).

Key Generation(PK, MSK, S, W_{ER_u})**:** This algorithm is done by the KGAs. It takes PK, MSK as input for both authorities. However, for KGA_2, it takes another input parameter which is a set of authorized location areas with their corresponding associated areas, S where $|S| \leq \mathbb{L}$ for an ER_u or a group of ERs, and for KGA_1 it takes the set of attributes of the user W_{ER_u}. KGA_1 picks $r'_u \in_R \mathbb{Z}_N$, and KGA_2 picks $r_u \in_R \mathbb{Z}_N$. Finally, they output the user's secret key as in (5.3).

$$SK_u \leftarrow \Big\{SK_0 = H_1(ER_u)^{\beta}, SK_1 = q^{\alpha}\Lambda^{r'_u}, SK_2 = g_1^{-r'_u}, SK_3 = g_2^{-r_u},$$
$$SK_4 = R^x h_0^{r_u} g_2^{\alpha'}, SK_{i,j} = T_{i,j}^{r'_u} \qquad \forall v_{i,j} \in W_{ER_u},$$
$$SK'_{j_1} = h_{j_1}^{r_u} \qquad \forall j_1 \in \{n\} \setminus \{S\}, SK''_{j_2} = h_{j_2}^{r_u} \qquad \forall j_2 \in \{S\}\Big\}, \tag{5.3}$$

and gives SK_u to the ER_u. Note that SK_0, SK_4, and SK''_{j_2} will be securely transferred to the GPS component of the user's smartphone. The GPS also receives S.

Encrypt(PK, M, D, S')**:** A DO chooses $s, t \in_R \mathbb{Z}_N$, $D = \{D_1, D_2, ..., D_{l'}\}$ as an access structure where $D_i \in V_i$ and $|l'| \leq |l|$, $S' = L_c \bigcup S''$ where L_c is the current location (if $L_c \in S''$ then $S' = S''$), the message M (which is comprised of the DO's health record, emergency information, etc.), and computes (5.4) and (5.5).

$$K = Y_0^s \times Y_1^t \tag{5.4}$$

$$C \leftarrow \Big\{D, S', C_0 = \xi'_K(M), C_1 = g_1^s, C_2 = g_2^t, C_3 = (\Lambda \prod_{v_{i,j} \in D} T_{i,j})^s \times (h_0 \prod_{j \in S'} h_j)^t\Big\} \tag{5.5}$$

where ξ' is a symmetric encryption scheme (e.g. AES). The DO sends C to the CS. Note that for all distinct access structures $\forall D, D'$, $\sum_{v_{i,j} \in D} t_{i,j} \neq \sum_{v_{i,j} \in D'} t_{i,j}$ is assumed.

Key-agreement(L_E, CS_{id}, r)**:** This algorithm is done by the GPS component of an ER's device. It takes as input the location of an emergency L_E, the identity of the CS and $r \in_R \mathbb{Z}_N$. GPS checks if the location of the ER is in AA_{L_E}. If the check passes, it generates a pseudonym $\theta = Q_{ER_u}^r$, and a one-way session key as in (5.6).

$$k = e(Q_{ER_u}, Q_{CS})^{r\beta} \tag{5.6}$$

GPS sends back $\{\theta||\xi'_k(L_{id}||E'_t||\eta)\}$ to the ER where $\eta \in_R \mathbb{Z}_N$ is a random nonce and E'_t is the current time. The GPS stores k, L_{id}, η, E'_t.

CS-Encrypt$(C_1, L_{id}, \tau_t, PK, x)$**:** The CS receives an emergency trigger message from PSAP as $E_{Trigger} = \{L_E||E_t||\tau_t\}$ where E_t is the emergency occurrence time, and τ_t is the time interval within which ERs' data requests are valid. The CS also receives a request from an ER consisting of $\{\theta||\xi'_k(L_{id}||E'_t||\eta)\}$ from which the server uses θ to generate the shared key k and decrypts $\xi'_k(L_{id}||E'_t||\eta)$. The CS checks if $E'_t \in \tau_t$ and retrieves data. Note that time intervals could be defined by the authorities considering the maximum response time of ERs. Afterwards, the CS checks whether $L_{id} \bigcap S' \neq \emptyset$. If the check passes, the CS modifies C_2 in order to incorporate the time attribute to the ciphertext as

$$C'_2 = g_2^t \times R^{1/(x+H_k(L_{id}||\tau_t||\eta))}. \tag{5.7}$$

The CS sends back to the ER the new ciphertext $C_{new} = (S', D, C_0, C_1, C'_2, C_3)$.

GPS-KGen(SK_4, PK, S', τ_t)**:** This algorithm is done by the GPS component. The user sends S' and τ_t to the GPS. The GPS will check whether $S \bigcap S' \neq \emptyset$ and $E'_t \in \tau_t$. If the checks pass, the GPS picks the corresponding $h_j^{r_i}, \forall j \in S \bigcap S'$, and generates a one-time key.

$$SK'_4 = R^{H_k(L_{id}||\tau_t||\eta)} \times R^x h_0^{r_u} g_2^{\alpha'} \prod_{j \in \{S \bigcap S'\}} h_j^{r_u}, \tag{5.8}$$

and sends it to the user. Note that in the time interval τ_t, the shared key k is valid. Here, for every new S', the GPS generates a new $\eta_{new} = H_k(\eta_{old})$. This changes the value of SK'_4 for the new S'.

Decrypt(PK, C, SK_u, S', S, D)**:** The user extracts K as follows,

$$K = \frac{K' \times K''}{e(R, R)}, where \tag{5.9}$$

$$K' = e\Big(C_1, SK_1 \times \prod_{v_{i,j} \in D} SK_{i,j}\Big) \times e\Big(C_3, SK_2\Big), \tag{5.10}$$

$$K'' = e\Big(SK'_4 \times \prod_{j_1 \in S'} SK'_{j_1}, C'_2\Big) \times e\Big(SK_3, C_3\Big). \tag{5.11}$$

Correctness: We check that decryption recovers the correct value of K,

$$\begin{aligned} K' =& e\Big(g_1^s, q^\alpha \Lambda^{r'_u} \times \prod_{v_{i,j} \in D} T_{i,j}^{r'_u}\Big) \times e\Big((\Lambda \prod_{v_{i,j} \in D} T_{i,j})^s \times (h_0 \prod_{j \in S'} h_j)^t, g_1^{-r'_u}\Big) \\ =& e\Big(g_1^s, q^\alpha\Big) \times e\Big(g_1, \Lambda \times \prod_{v_{i,j} \in D} T_{i,j}\Big)^{sr'_u} \times \Big(\Lambda \prod_{v_{i,j \in D}} T_{i,j}, g_1\Big)^{-sr'_u} \\ =& e(g_1, q)^{s\alpha}. \end{aligned} \tag{5.12}$$

And,

$$\begin{aligned} SK'_4 \times \prod_{j_1 \in S'} SK'_{j_1} =& R^{H_k(L_{id}||\tau_t||\eta)} \times R^x h_0^{r_u} g_2^{\alpha'} \prod_{j \in \{S \bigcap S'\}} h_j^{r_u} \times \prod_{j_1 \in S' \setminus \{S \bigcap S'\}} h_{j_1}^{r_u} \\ =& R^{(H_k(L_{id}||\tau_t||\eta)+x)} g_2^{\alpha'} (h_0 \prod_{j \in S'} h_j)^{r_u}. \end{aligned} \tag{5.13}$$

Plugging (5.13) into (5.11) gives us

$$\begin{aligned}
K'' =& e\Big(R^{(H_k(L_{id}||\tau_t||\eta)+x)}, R^{1/(x+H_k(L_{id}||\tau_t||\eta))}\Big) \times \\
& e(g_2^{\alpha'}, g_2^t) \times e\Big((h_0 \prod_{j \in S'} h_j)^{r_u}, g_2^t\Big) \times \\
& e\Big(g_2^{-r_u}, (\Lambda \prod_{v_{i,j} \in D} T_{i,j})^s \times (h_0 \prod_{j \in S'} h_j)^t\Big) \\
=& e(R, R) \times e(g_2, g_2)^{t\alpha'} \times e((h_0 \prod_{j \in S'} h_j), g_2)^{tr_u - tr_u} \\
=& e(R, R) \times e(g_2, g_2)^{t\alpha'}, \qquad (5.14)
\end{aligned}$$

Plugging (5.12) and (5.14) into (5.9) results in (5.15) as required.

$$K = Y_0^s \times Y_1^t \qquad (5.15)$$

Fig 5.3 illustrates the message (shown by arrows) exchange paradigm among ER_u, GPS, CS, and PSAP. Note that messages exchanged from 2 to 4 are assumed to be in the same time interval τ_t. As Fig 5.3 illustrates, an ER and the CS receive their corresponding emergency trigger message. The ER gets its location from the GPS component (messages 1-2), and forms a query to retrieve data (message 3). Note that this query should also contain the information that points to data belonging to a DO (e.g., a pseudonym UID_{DO}). Note that this information is merely a pseudonym and does not violate the privacy of a DO. This information is obtained using our proposed scheme in Chapter 4. After receiving the data response (message 4), the ER extracts S', τ_t and sends them to the GPS component (message 5). It generates the one-time key SK_4' and sends it back to the ER (message 6). Finally, the ER decrypts the ciphertext.

5.5.2 Updating Ciphertext

In our construction, the purpose of updating ciphertext is to change the access policy. An access policy is comprised of a subset of locations and a subset of static attributes.

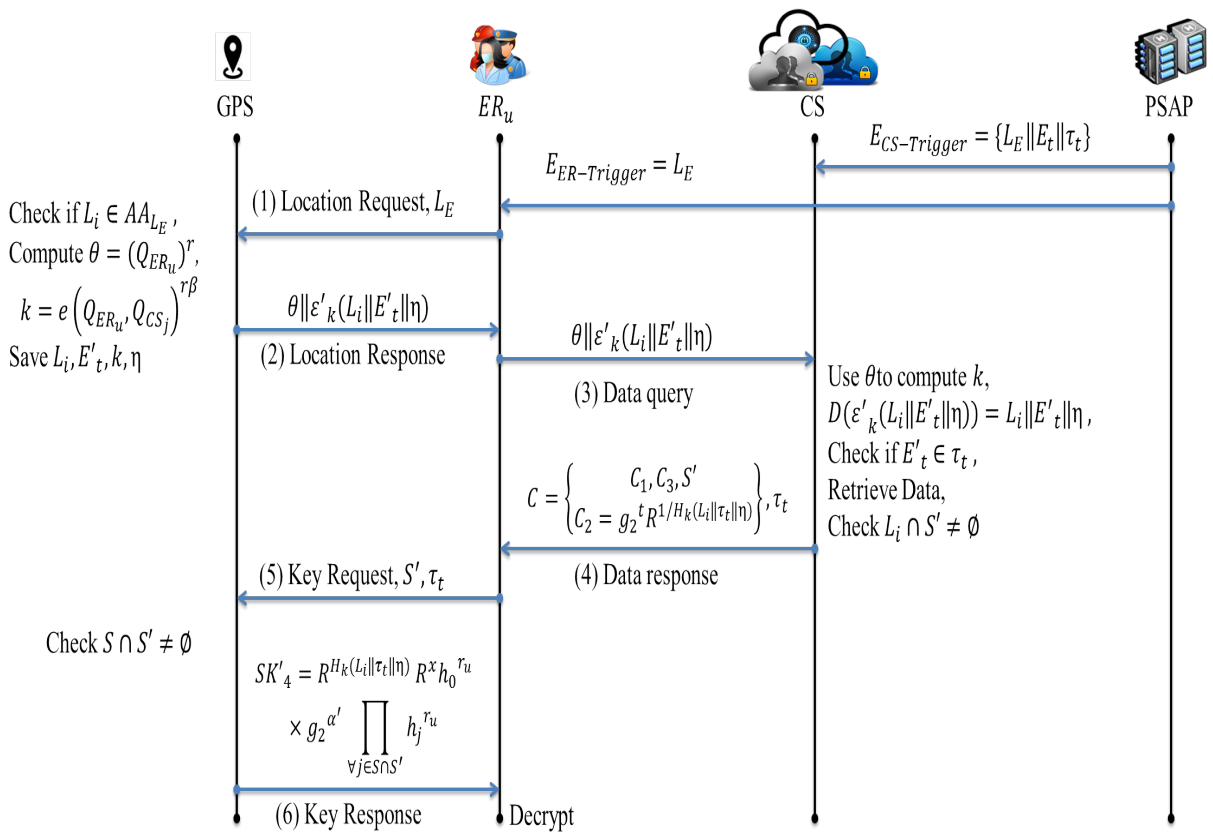

Figure 5.3: Message exchange paradigm

Alteration in either one of the subsets, changes access privileges. Note that DOs have control over their information and in defining and updating the authorization set. However, this task can be delegated to their smartphones to automate the process. In this regard, if a DO moves out of his preferred location set, it will update the ciphertext.

For example, if a DO defines an access policy as $D = \{v_{2,3}, v_{4,2}, ..., v_{6,3}\}$, and he wants to remove $v_{4,2}$ and add $v_{8,1}$, the updating massage will be $C_{updating} = \{D_{new}, T_{4,2}^{-s} \times T_{8,1}^{s}\}$ which will be sent to the CS. Then, the CS modifies the corresponding record as follows: $C_{3,new} = C_3 \times T_{4,2}^{-s} \times T_{8,1}^{s}$. Note that changing the location set is done following the same procedure.

5.5.3 Outsourcing Partial Decryption

Outsourcing decryption has been used in many works to transfer the heavy burden of pairing computation to one or several powerful servers. However, it is important that the server does not learn any information about the private key or the message during the partial decryption process. Notice that the decryption algorithm requires D and S' in order to proceed. Here, those sets can be acquired using a round of communication between an ER_u and the CS. Consider the following suppositions:

$$SK_1 \times \prod_{v_{i,j} \in D} SK_{i,j} = A, \tag{5.16}$$

$$SK_2 = B, \tag{5.17}$$

$$SK_3 = C, \tag{5.18}$$

$$SK'_4 \times \prod_{j_1 \in S'} SK'_{j_1} = D. \tag{5.19}$$

Then, the ER chooses a random number $\delta \in_R \mathbb{Z}_N$ and instead of message (3) in Fig 5.3 sends $\{A^\delta || B^\delta || C^\delta || D^\delta || L_{ER_u} || \theta || \xi'(\tau_t || \eta) || UID_{DO}\}$ to the CS. The server computes the following

$$K'_{new} = e(C_1, A^\delta) \times e(C_3, B^\delta) = K'^\delta, \tag{5.20}$$

and,

$$K''_{new} = e(D^\delta, C'_2) \times e(C^\delta, C_3) = K''^\delta, \tag{5.21}$$

and sends $K'_{new} \times K''_{new}$ back to the ER. The user computes (5.22) and decrypts the message.

$$K = \frac{(K'_{new} \times K''_{new})^{1/\delta}}{e(R, R)} \tag{5.22}$$

5.6 Security Analysis of LA-CP-ABE

In this section, we will analyze the security of the LA-CP-ABE scheme. We will first discuss some security points of the proposed scheme. Then, we prove that the proposed authorization scheme is selectively secure under the $m-$BDHE assumption.

The ER and an observer of the message (3) in Fig 5.3 cannot undetectably manipulate $\xi'_k(L_{id}||\tau_t||\eta)$ since it is protected using the AES symmetric encryption scheme. The symmetric shared key is computed using SK_0 and a random number. Recall that SK_0 was securely transferred to the GPS. Moreover, the random number is changed for every location query which causes the shared key to change accordingly. Therefore, an ER cannot bypass a GPS to generate legitimate data queries himself/herself. In addition, the CS ensures that the message has been generated by GPS and proceeds with the algorithm.

In addition, the secure GPS of a smartphone computes the session private key SK'_4 by which the message can be decrypted. Note that only if the location and time interval attributes of the ciphertext match the ones in the SK'_4, will the $e(R, R)$ component be eliminated. Otherwise, it would have some unknown exponent which causes the decryption to fail. In this case, such data is filtered and considered irrelevant to the ongoing situation.

Furthermore, an ER should not be able to extract the private key element SK_4 of its GPS to be able to bypass it. To mitigate that, a new unique $\eta_{new} = H_k(\eta_{old})$ is computed for each new set of locations S' (message (5) in Figure 5.3). In this case, the queries from the ER to the GPS result in random looking varied values. Figure 5.4 illustrates key query/response when $S' \subset S$ is assumed and the queries are in the same time interval and at the same location (i.e., k is fixed). Even under such assumptions, the ER is not able to extract $X = R^x h_0^{r_u} g_2^{\alpha'}$ or $h_j \in Q_i \setminus Q_{i'}$. In Figure 5.4, $h_3^{r_u}$ or $h_2^{r_u}$ cannot be extracted from $\frac{SK'_{4,2}}{SK'_{4,1}}$ or $\frac{SK'_{4,1}}{SK'_{4,3}}$ respectively since η is changing for each new Q_i; therefore, $R^{H_k(L_{id}||\tau_t||\eta)}$ is changing for respective queries. Notice that the GPS cannot extract $g_2^{\alpha'}$ since it has been blinded using R^x. And, since they are orthogonal to each other (5.1), this obfuscating element will be cancelled in the pairing computation in the decryption process.

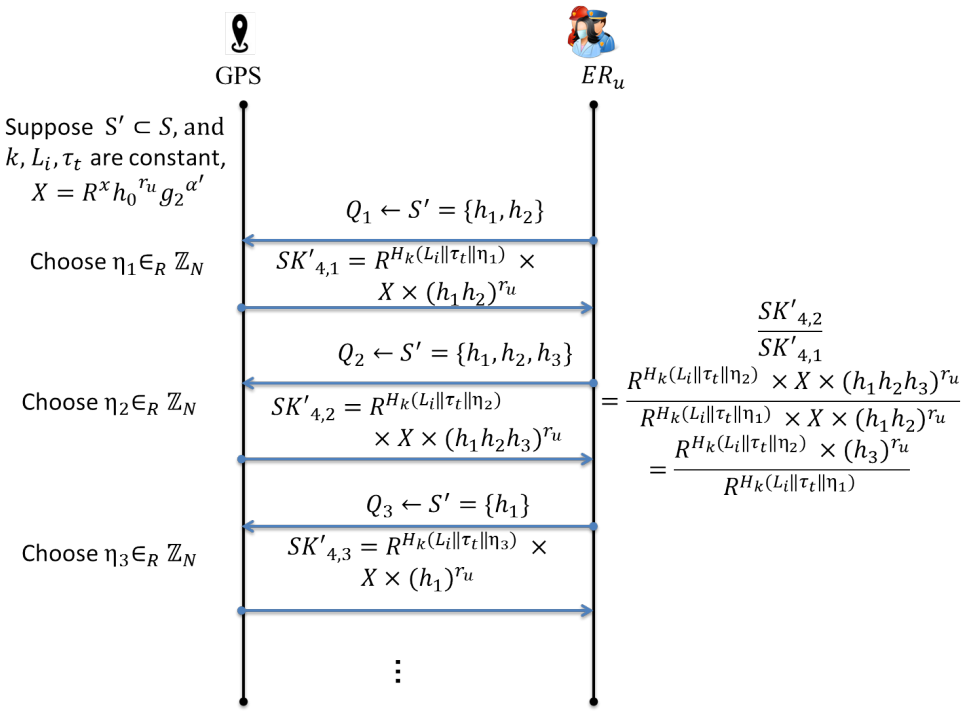

Figure 5.4: Key query/response between an ER and GPS

The proposed scheme calculates a session key from the term θ to prevent a replay attack. In this regard, the decryption key is valid for a certain location area and within a single time interval. This also is an important factor to consider for addressing our new threat model. For example, if an ER receives a private key for some data associated with a certain location area and time interval τ_t, then the user should not be able to decrypt the data for the same location within another time interval $\tau_{t+t'}$. In this case, our access model is superior to the Break-the-Glass model. Recall that in that concept, the master key always decrypts the corresponding ciphertext. Our scheme restricts data access based on the time and location of an emergency by which irrelevant data is filtered and a higher level of privacy is provided. Besides, considering the proposed scheme, both the CS and DOs are involved in the authorization process. This decreases the risk of privacy breach if the server turns malicious.

We integrated BE with CP-ABE to incorporate the location attribute in an access policy. In this case, using BE implies a $(1, n)-$threshold access structure meaning that possessing private keys corresponding merely to one location attribute leads to successful

decryption of the message. Observe that if a DO has included his/her home and work location areas (e.g. L_1, L_2 respectively) in the ciphertext, an ER who is visiting either one of the Associated areas (e.g. AA_{L_1}) can generate the proper private key to decrypt his/her data even if the DO is currently located at the other location (L_2 in this example). This can be avoided by updating the ciphertext by the DO upon leaving the home location area. Here, there is a trade-off between the ciphertext updating computation/communication costs and data filtering accuracy: the bigger the size of S'', the smaller the data filtering accuracy. We will further demonstrate such costs in Section 5.7.

In addition, to decrease the probability of privacy breach, we can incorporate an Audit-trail technique to log all of the activities, which can be used to spot unauthorized data access [155]. Here, this process should be operated and managed by a trusted party to avoid collusion or illegitimate modifications to the log.

Moreover, the CS cannot learn anything from the outsourcing computations since the components are blinded by the exponent δ. Finally, we also avoid the key escrow problem, since we use two separate KGAs to generate secret keys of the system. Note that setup algorithms in both KGAs receive the same description of $(\mathbb{G}, \mathbb{G}_T, e)$. One of the KGAs generates the private keys corresponding to location attribute, and the other one for the static attributes. Therefore, there is no single key generating authority that can decrypt all messages. Here, we assume that KGAs do not collude with each other.

We prove our LA-CP-ABE scheme is selectively secure based on the decisional $m - BDHE$ assumption. Note that if BDHE is assumed to be hard in the subgroups G_{p_1} and G_{p_2}, then it can be assumed to be hard in the composite order group $\mathbb{G}$ as well. We follow the selective CPA security game in which we assume that an attacker $\mathcal{A}$ wins the game with advantage ϵ. (It is worth mentioning that we can also achieve CCA security as well using well-known methods used in [156, 157].) We construct an algorithm $\mathcal{B}$ that uses $\mathcal{A}$ to break the $m - BDHE$ assumption with the advantage of at least ϵ. The $m - BDHE$ challenger generates two problem instances as below.

For the subgroup G_{p_1}, the instance is comprised of g_1^c and the set (5.23) where $m = l \times |V_i| + |ERs|$ is the number of attributes times the size of their value set plus the number

of ERs in the system. Suppose $|V_i| = v$ is the same for all of the attributes. For the subgroup G_{p_2}, the instance is comprised of $g_2^{c'}$ and the set (5.24) where $m' = n + |ERs|$ is the size the location set in the system plus the number of ERs in the system.

$$(g_1, g_1^a, g_1^{(a^2)}, ..., g_1^{(a^m)}, g_1^{(a^{m+2})}, ..., g_1^{(a^{2m})}, Z_1) \in \mathbb{G}^{2m} \times \mathbb{G}_{T,p_1}, \tag{5.23}$$

$$(g_2, g_2^b, g_2^{(b^2)}, ..., g_2^{(b^{m'})}, g_2^{(b^{m'+2})}, ..., g_2^{(b^{2m'})}, Z_2) \in \mathbb{G}^{2m'} \times \mathbb{G}_{T,p_2} \tag{5.24}$$

Suppose that the challenger selects $\rho \in_R \{0,1\}$ and $\rho' \in_R \{0,1\}$ (the two selections are independent of one another). If $\rho = 0 = \rho'$, then $Z_1 = e(g_1, g_1)^{ca^{m+1}}$, and $Z_2 = e(g_2, g_2)^{c'b^{m'+1}}$. Otherwise, if $\rho = 1 = \rho'$, then Z_1 and Z_2 are random elements of $\mathbb{G}_{T,p_1}, \mathbb{G}_{T,p_2}$ respectively. The challenger gives the two $m - BDHE$ instances to $\mathcal{B}$. Consider the game between $\mathcal{B}$ and the adversary $\mathcal{A}$ as follows:

***Initialization*:** $\mathcal{A}$ commits to sets $D \subseteq [1, m - |ERs|]$ and $S' \subseteq [1, m' - |ERs|]$.

***Setup*:** $\mathcal{B}$ generates $\varphi = \{u_1, u_2, u_3, y_0, ..., y_m\}, \varphi' = \{u'_1, y'_0, ..., y'_{m'}\} \xleftarrow{R} \mathbb{Z}_N$. Note that we allocate $\{y_1, ..., y_v\}$ to A_1, $\{y_{v+1}, ..., y_{2v}\}$ to A_2, etc. In other words, suppose there is map function $\rho(i,j) = (i-1)v + j \in [1, m - |ERs|]$ for $1 \leq i \leq l$ and $1 \leq j \leq v$. This function is used to assign a number in $[1, m - |ERs|]$ to $T_{i,j}$. Then, $\mathcal{B}$ sets

$$q \leftarrow g_1^{u_1} \tag{5.25}$$

$$R \leftarrow g_1^{u_2} \tag{5.26}$$

$$\Lambda \leftarrow g_1^{y_0 u_3} \tag{5.27}$$

$$T_{i,j} \leftarrow g_1^{y_{\rho(i,j)}} \quad for \quad \rho(i,j) \in D \tag{5.28}$$

$$T_{i,j} \leftarrow g_1^{y_{\rho(i,j)} + a^{\rho(i,j)}} for \quad \rho(i,j) \in [1, m - |ERs|] \setminus D \tag{5.29}$$

$$h_0 \leftarrow g_2^{y'_0} \tag{5.30}$$

$$h_i \leftarrow g_2^{y'_i} \quad for \quad i \in S' \tag{5.31}$$

$$h_i \leftarrow g_2^{y'_i + b^i} \quad for \quad i \in [1, m'] \setminus S' \tag{5.32}$$

Note that $A_0 = \Lambda \in D$ is always true. Formally, $\mathcal{B}$ sets $\alpha \leftarrow y_0 u_3 \times a^{m+1}$ and $\alpha' \leftarrow y_0' u_1' \times b^{m'+1}$. Thus, public parameters are

$$PK = \Big\{ g_1, q, R, \Lambda, e(g_1,q)^{\alpha}, e(R,R), \{T_{i,j}\}_{i\in[1,l],j\in[1,v]}, g_2, e(g_2,g_2)^{\alpha'}, h_0, h_1, ..., h_n, \Big\} \quad (5.33)$$

where $e(g_1,q)^{\alpha}$ and $e(g_2,g_2)^{\alpha'}$ can be computed as

$$e(g_1,q)^{\alpha} = e(g_1^{a^m}, (g_1^a)^{u_1})^{y_0 u_3} = e(g_1, g_1^{u_1})^{y_0 u_3 \times a^{m+1}}, \quad (5.34)$$

and,

$$e(g_2,g_2)^{\alpha'} = e(g_2^{b^{m'}}, g_2^b)^{u_1'} = e(g_2,g_2)^{u_1' \times b^{m'+1}}. \quad (5.35)$$

$\mathcal{B}$ sends PK to $\mathcal{A}$.

***Private Key Queries*:** $\mathcal{A}$ is allowed to query the private key only for the attributes that were not included in either D or S' except Λ and h_0. We first generate the keys associated with $\mathbb{G}^{2m} \times \mathbb{G}_{T,p_1}$, and then for the other instance. $\mathcal{B}$ generates $z_u' \xleftarrow{R} \mathbb{Z}_N$ and formally sets $r_u' = z_u' - u_1 a^{m+1-u}$. For each ER, we personalize the default attributes as follows: $\Lambda_u = g_1^{y_0 u_3 a^u} = (g_1^{a^u})^{y_0 u_3}$, $h_{u,0} = g_2^{y_0' b^u} = (g_2^{b^u})^{y_0'}$. It outputs,

$$\begin{aligned} SK_u = \Big\{ & SK_0 = H_1(ER_u)^{\beta}, \quad (5.36) \\ & SK_{u,1} = q^{\alpha} \Lambda_u^{r_u'} = g_1^{u_1 u_3 y_0 a^{m+1}} \times g_1^{z_u' u_3 y_0 a^u - u_1 u_3 y_0 a^{m+1} a^{u-u}} = g_1^{z_u' u_3 y_0 a^u}, \\ & SK_{u,\rho(i,j)} = T_{i,j}^{r_u'} = g_1^{(y_{\rho(i,j)} + a^{\rho(i,j)}) r_u'} \\ & = g_1^{(y_{\rho(i,j)} + a^{\rho(i,j)})(z_u' - u_1 u_3 y_0 a^{m+1-u})} for \quad \rho(i,j) \in [1, m - |ERs|] \setminus D, \\ & SK_{u,2} = g_1^{-r_u'} \Big\}. \end{aligned}$$

For the instance associated with $\mathbb{G}^{2m'} \times \mathbb{G}_{T,p_2}$, $\mathcal{B}$ generates $z_u, x \xleftarrow{R} \mathbb{Z}_N$ and formally sets $r_u = z_u - u'_1 b^{m'+1-u}$. It outputs the corresponding elements of the secret key as follows

$$
\begin{aligned}
SK_u = \Big\{ & SK_{u,3} = g_2^{-r_u}, \\
& SK'_{u,j_1} = h_{j_1}^{r_u} = g_2^{r_u(y'_{j_1}+b^{j_1})} \quad for \quad j_1 \in \{n\} \setminus \{S \bigcup S^*\}, \\
& SK_{u,4} = R^x h_{u,0}^{r_u} g_2^{\alpha'} = g_1^{u_2 x} \times g_2^{z_u y'_0 b^u - y'_0 u'_1 b^{m'+1} b^{u-u}} \times g_2^{y'_0 u'_1 b^{m'+1}} = g_1^{u_2 x} g_2^{z_u y'_0 b^u}, \\
& SK''_{j_2} = h_{j_2}^{r_u} = g_2^{r_u(y'_{j_2}+b^{j_2})} \quad for \quad j_2 \in \{S\} \setminus \{S^*\} \Big\}
\end{aligned} \tag{5.37}
$$

Remarks: The tricky part is to simulate the $SK_{u,1}$ and $SK_{u,4}$ values since they include terms of the form $g_1^{a^{m+1}}$ and $g_2^{b^{m'+1}}$ respectively. These terms are unknown to $\mathcal{B}$. However, notice that these terms in the exponent are cancelled out which makes $SK_{u,1}$ and $SK_{u,4}$ computable for $\mathcal{B}$. In addition, the distribution of the private key is identical to that of the original scheme.

***Challenge*:** $\mathcal{A}$ chooses a subset $D^* \subset D, S^* \subset S'$, two messages M_0, M_1 and sends them to $\mathcal{B}$. $\mathcal{B}$ chooses $\mu \in_R \{0,1\}$, computes the ciphertext as below, and sends the result to $\mathcal{A}$.

$$
K = Z_1^{u_1 u_3 y_0} \times Z_2^{y'_0 u'_1} \tag{5.38}
$$

$$
\begin{aligned}
C^* = \Big\{ & C_0^* = \xi'_K(M_\mu), C_1^* = g_1^c, C_2^* = g_2^{c'}, \\
& C_3^* = (\Lambda \prod_{v_{i,j} \in D^*} T_{i,j})^c \times (h_0 \prod_{j \in S^*} h_j)^{c'} = (g_1^{y_0 u_3} \prod_{v_{i,j} \in D^*} g_1^{y_{\rho(i,j)}})^c \times (g_2^{y'_0} \prod_{j \in S^*} g_2^{y'_j})^{c'} \\
& = (g_1^{c y_0 u_3} (g_1^c)^{\sum_{v_{i,j} \in D^*} y_{\rho(i,j)}}) \times (g_2^{c' y'_0} (g_2^{c'})^{\sum_{j \in S^*} y'_j}). \Big\}
\end{aligned} \tag{5.39}
$$

Notice that $\mathcal{B}$ is able to calculate the challenge from the instances as shown above.

***CS-encrypt, GPS-KGen*:** We use the random oracle model instantiated with HMAC to output the required randomness for the two algorithms.

***Guess*:** Finally, $\mathcal{A}$ outputs a bit $\mu' \in \{0,1\}$. $\mathcal{B}$ outputs 1 if $\mu = \mu'$ and 0 otherwise. Notice that if $Z_1 = e(g_1, g_1)^{ca^{m+1}}$ and $Z_2 = e(g_2, g_2)^{c'b^{m'+1}}$, then C^* is a valid challenge ciphertext associated with D^*, S^*. Therefore, $\mathcal{A}$ has advantage ϵ. Since $m-$BDHE is known to be a hard problem, the advantage ϵ is negligible. Then, we have the following,

$$\Pr[\mathcal{B} \rightarrow 1 | Z_1 = e(g_1, g_1)^{ca^{m+1}}, Z_2 = e(g_2, g_2)^{c'b^{m'+1}}] = \tag{5.40}$$
$$\Pr[\mu = \mu' | Z_1 = e(g_1, g_1)^{ca^{m+1}}, Z_2 = e(g_2, g_2)^{c'b^{m'+1}}] = \frac{1}{2} + \epsilon.$$

Otherwise, three other cases may occur; first, both $Z_1 \in \mathbb{G}_{T,p_1}$ and $Z_2 \in \mathbb{G}_{T,p_2}$ are random elements; second and third, either one of them is a random element. In all of those cases, $\mathcal{A}$ has no advantage to distinguish the ciphertext generated for M_0 from the ciphertext generated for M_1. This is because all parts of the ciphertext have the same distribution in either $\mu = 0$ or $\mu = 1$. Therefore, $\Pr[\mathcal{B} \rightarrow 0 | Z_1$ and Z_2 are $random] = \Pr[\mathcal{B} \rightarrow 0 | Z_1$ or Z_2 are $random] = \frac{1}{2}$. $\square$

Note that since x and $u_2 \in \mathbb{Z}_N$ are chosen uniformly at random, and $g_1 \in G_{p_1}$, then, $g_1^{u_2 x} = R^x$ reveals nothing about the value of $u_2 x$ modulo r. In other words, $u_2 x$ modulo r is uniformly random. Therefore, in the view of an attacker, the corresponding key is well-distributed. Another underlying assumption is that breaking the symmetric cipher (e.g., AES) is intractable.

5.7 Performance Analysis of LA-CP-ABE

In this section, we discuss some significant features of our LA-CP-ABE scheme with regards to the emergency response application. In addition, we analyze computation and communication complexities, storage requirements, and delay. Concerning with CP-ABE schemes, our focus is merely on the constant ciphertext-size and a constant number of pairings in the decryption process.

Table 5.1: CP-ABE protocol comparison

Scheme	Complexity	Ciphertext Size	Decryption Computation overhead	Key Generation computation	Private key size	Access structure	CT-update communication	CT-update computation	Dynamic attribute
[110]	DBDH	$2\|G_1\|+\|G_T\|+\|AS\|$	$3\tau_p+2G_T$	$(W+4)G_1$	$2\|G\|$	(n_{att},n_{att})–Threshold	$\|G\|$	$(i+j)G$	X
[111]	Composite-Order Bilinear Group, DBDH	$\|G_T\|+2\|G\|+\|AS\|$	$3\tau_p+2G_T$	$(W+3)G_p$	$2\|G\|$	(n_{att},n_{att})–Threshold	$\|G\|$	$(i+j)G$	X
[106]	Composite-Order Bilinear Group	$\|G_T\|+2\|G\|$	$2\tau_p+(2W)G+2G_T$	$(l_{AS}-t_{sh}+1)(n_{att}+2)G$	$(l_{AS}-t_{sh}+1)(n_{att}+2)\|G\|$	(t_{sh},n_{att})–Threshold	$\|G\|$	$(i+j)G$	X
[112]	aMSE-DDH	$\|G_1\|+\|G_2\|+\|G_T\|+\|AS\|$	$3\tau_p+t_{sh}G_1+(n_{att}+t_{sh}-2)G_T$	$WG_1+(n_{att}-1)G_2$	$(n_{att}+\|W\|)\|G\|$	(t_{sh},n_{att})–Threshold	$\|G\|$	$(i+j)G$	X
[113]	DBDH	$\|G_T\|+2\|G\|+\|AS\|$	$2\tau_p+2G_T$	$(2W+4)G$	$2\|G\|$	(n_{att},n_{att})–Threshold	$\|G\|$	$(i+j)G$	X
[158]	m-BDHE	$2\|G\|+\|\xi'\|$	$(2n_{att}+1)\tau_p$	$(2n_{att}+1)G$	$(2n_{att}+1)G$	AND_{+-*}	$\|G\|$	$(i+j)G$	X
[159]	B-Co-CDH	$3\|G\|$	$3\tau_p+G$	$5G$	$4\|G\|$	(n_{att},n_{att})–Threshold	X	X	X
[157]	m-BDHE	$\|G_T\|+2\|G\|+\|AS\|$	$2\tau_p+2G_T$	$2n_{att}G$	$n_{att}\|G\|+\mathbb{Z}_p$	AND_{+-}	$2\|G\|$	$2(i+j)G$	X
[160]	(t_{sh},ϵ,l)–BDHE	$2\|G\|+\|G_T\|$	$2\tau_p+2G_T$	$3n_{att}G$	$n_{att}\|G\|+\mathbb{Z}_p$	AND_{m*}	$2\|G\|$	$2(i+j)G$	X
$[161]_1$ $[161]_2$	m-BDHE	$\{\|G_T\|+2\|G\|\}$ $\{\|G_T\|+3\|G\|+\|\mathbb{Z}_p\|\}$	$\{2\tau_p+(2n_{att})G\}$ $\{6\tau_p+(2n_{att}+2)G\}$	$(W+d_{def})(2n_{att}+2)G$	$(n_{att}+\|W\|)(2n_{att}+1)\|G\|$	(t_{sh},n_{att})–Threshold	$\|G\|$	$(i+j)G$	X
[162]	Composite-Order Bilinear Group	$\|G_T\|+2\|G\|$	$3\tau_p$	$(W+3)G$	$2\|G\|$	(n_{att},n_{att})–Threshold	$\|G\|$	$(i+j)G$	X
[163]	m-BDHE	$\{\|G_T\|+3\|G\|+\|\mathbb{Z}_p\|\}$	$\{6\tau_p+(2n_{att}+2)G\}$	$(W+d_{def})(2n_{att}+2)G$	$(n_{att}+\|W\|)(2n_{att}+1)\|G\|$	(t_{sh},n_{att})–Threshold	$\|G\|$	$(i+j)G$	X
Ours	m-BDHE	$3\|G\|+\|\xi'\|$	$(4+\|S'+D\|)G+2G_T$	$(7+\|S+W_{ER}\|)G$	$(6+\|S+W_{ER}\|)\|G\|$	AND_m	$\|G\|$	$(i+j)G$	✓

Table 5.1 presents a comprehensive comparison among state-of-the-art CP-ABE protocols. In this table, n_{att} is the number of attributes in the universe, l_{AS} is the number of attributes included in an access policy, t_{sh} is the threshold value for which an access policy will be satisfied, v is the number of values associated with an attribute, W is the number of attributes that a user possesses, d_{def} number of default attributes in the system, $\text{AND}_+ - *$ means AND gate with positive, negative, and wildcards, AND_m* means AND gate with multivalues and wildcards, and i, j are the numbers of added and removed attributes respectively to the ciphertext.

In Table 5.1, ciphertext size is constant in all of the schemes regardless of the number of attributes in the access policy. In addition, the table shows that the ciphertext contains an element of the target group $\mathbb{G}_T$ for all schemes except the works [158, 159] and ours. We used such an element as the secret key to a symmetric encryption scheme ξ' (e.g., AES) to increase computation efficiency.

Besides, based on Table 5.1, computation complexity for all of the schemes is constant in terms of the number of pairing computations. In this case, our scheme is more efficient than others, since we delegated such heavy burden to a powerful server. Notice that our outsourced decryption scheme requires merely 4 pairing operations on the server side. On the other hand, our decryption computation complexity is comprised of $(4 + |S' + D|)$ multiplications in $\mathbb{G}$ and 1 exponentiation plus 1 multiplication in the target group $\mathbb{G}_T$ on the user side. This is important considering the emergency response application. It can be shown that the delay corresponding to pairing computations is much higher than group arithmetic operations. Table 5.2 indicates the difference in computation delay between bilinear group multiplication/pairing with 80-bit security and AES encryption with 128-bit security. The numbers were extracted from the works in [164, 165, 166]. The first three rows show the timing costs for resource constrained devices, and the last one illustrates a more powerful computer. Observe that even in resource constrained devices, as long as the computation complexity remains constant especially for pairing calculations, the respective schemes are still feasible. However, the advantage of our scheme is the use of AES instead of multiplication of the message with a key (e.g., $e(X, X)^x \in \mathbb{G}_T$) which results in higher

Table 5.2: Time costs comparison

	Platform	**Time**
Multiplication-80bit	MSP430 TelosB 8MHz	0.001ms
Pairing-80bit	MSP430 TelosB 8MHz	1.27s
AES-128	ATmega128 16MHz	160Kbit/s
AES-128	AMD64 2.194GHz	198Mbit/s

efficiency even on resource constrained devices.

Comparing the two columns for private key size and access structure in Table 5.1 shows an interesting conclusion which is the fact that constant private key size results in a very limited access structure $(n_{att}, n_{att})-$Threshold. In this case, although a user might have several attributes, there is only one combination that enables him to decrypt a ciphertext. In other words, the attributes and their values used in the ciphertext should perfectly match the ones in a user's private key. On the contrary, an AND-gate access structure provides a more flexible and expressive access policy at the cost of greater private key size. In this case, the storage requirement for a user demands higher capacity. Our scheme uses multivalued AND-gates.

In addition, Table 5.1 shows that all the schemes are able to update the ciphertext using the same technique as ours except [159]. Note that those schemes did not present the procedure with which a ciphertext can be updated. The scheme [159] requires to contact a server in order to get the hashed value of the new authorized attribute list and update the aggregated group element in the ciphertext. In fact, to update a ciphertext, that scheme substitutes two out of three elements of the ciphertext whereas others merely modify existing element(s) by multiplication as shown in Section 5.5.2. The cost of updating ciphertext is similar for all of the schemes except [157, 160] for which two elements of the ciphertext should be modified. Note that the communication cost of updating a ciphertext does not depend on the number of attributes. This is a key advantage especially in situations where users often change their locations.

Furthermore, the key distinction among the schemes in Table 5.1 is the ability to incorporate dynamic attributes (i.e., location and time). Our scheme uses BE to incorporate the location attribute to ciphertext at the DO's side, and a server incorporates the time

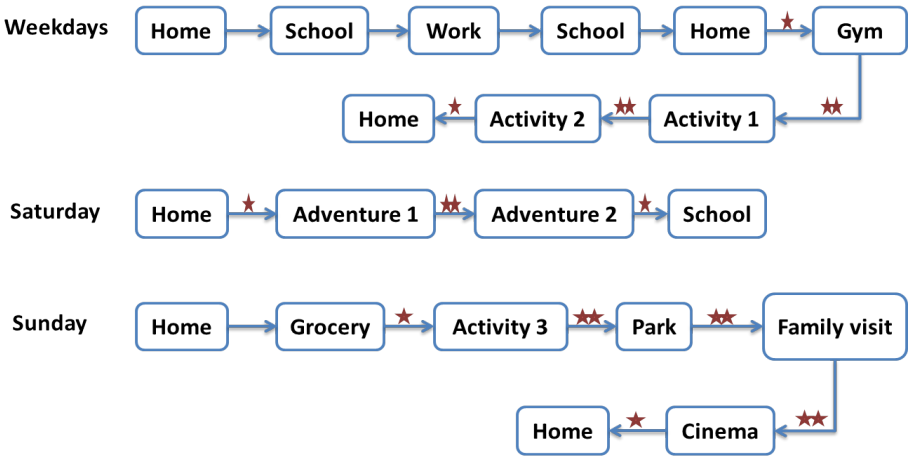

Figure 5.5: Movement trajectory scenario per week

attribute to complete the requirement. None of the other schemes in Table 5.1 is able to incorporate dynamic attributes.

Utilizing BE decreases the updating frequency which results in a higher degree of computation and communication efficiency. Suppose there are two individuals Alice and Bob who have the same life style meaning that their movements during a week is similar. Figure 5.5 illustrates such a scenario in which weekdays and the weekend are shown separately. In this figure, boxes are locations and it is assumed that each box is in a distinct location area. This implies that, for instance from Home to School, a DO needs to update the ciphertext.

Assume that the preferred location sets of Alice and Bob are $S''_{Alice} = \{Home,\ School,\ Work,\ Grocery\ store\}$ and $S''_{Bob} = \{\emptyset\}$ respectively. Therefore, if Alice moves from one location to another in her set, no updating is necessary. Note that in this case $S' = S''$. In addition, if she moves from a location in the set (e.g., Home) to another one not in the set (e.g., Gym), she only needs to update the ciphertext by adding Gym to it. The stars on top of the arrows in Figure 5.5 show when an updating is required for Alice. In addition, the number of stars shows the total number of group multiplications necessary for updating the ciphertext. Unlike Alice, Bob needs to update his data for each one of his movements.

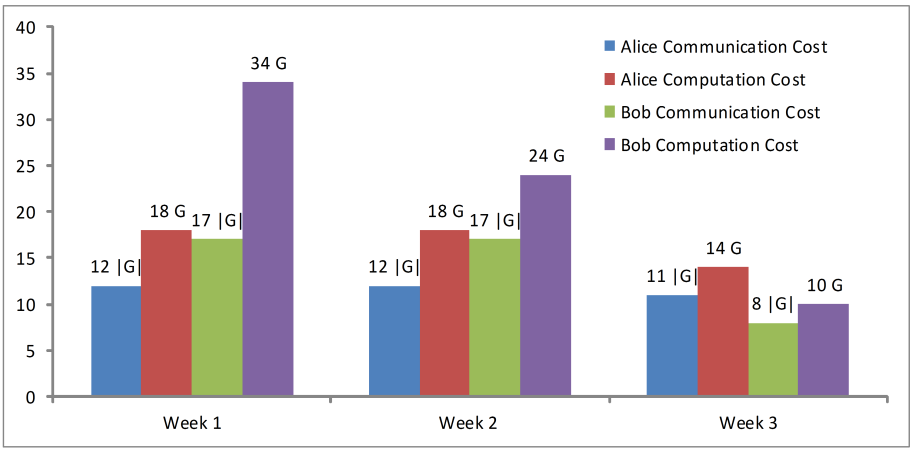

Figure 5.6: The effect of $|S''|$ on communication and computation costs of updating process

Figure 5.6 illustrates the communication and computation costs of the updating process. The horizontal axis shows weeks one through three for which we change S' (recall that $S' = S'' \bigcup L_c$, i.e., the union of the preferred location set and the current location) for both Alice and Bob. The first set of bars represents the scenario shown in Figure 5.5. The second set shows the effect of $|S''_{Bob}|$ when Bob includes *Home* and *Work* into his set. Finally, the third set of bars shows the costs when Alice includes *Gym* and *Park* into her set, and Bob includes all boxes except the three *Activities* and the two *Adventures*. Figure 5.6 shows that increasing $|S''|$ affects computation complexity more than communication overhead. In this case, the computation complexity and communication overhead of Bob decreased by approximately 60 and 52 percent respectively per week comparing the first week with the second and the third respectively. And, Alice was able to decrease her computation cost by 32 percent and her communication overhead by 8.3 percent per week from week two to three. It can be concluded that if $|S''|$ increases, the ciphertext updating costs (i.e., communication and computation costs) decrease. However, it also decreases the data filtering accuracy. The relationship between data filtering accuracy and $|S'|$, and n is shown in (5.41) (for $S' = S''$). Here, $1 \leq |S'| \leq n$ which indicates that the accuracy is 100% when $|S'| = 1$ and it drops to 0% when $|S'| = n$.

$$Accuracy = \frac{n - |S'|}{n - 1} \times 100 \tag{5.41}$$

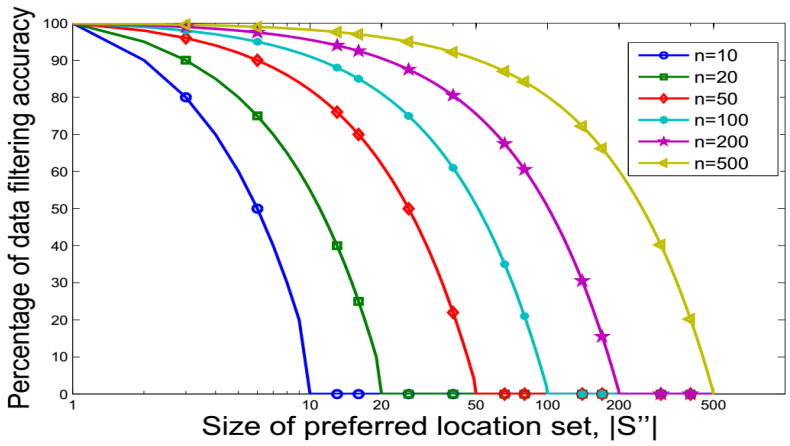

Figure 5.7: Percentage of data filtering accuracy with regards to total number of location areas n

Table 5.3: Comparison of total computation and communication overhead

Scheme	Communication Overhead				Decryption computation Cost	
	Ciphertext size	Outsourcing communication	Public parameter	Secret key size	User	Outsourced
[109]	$\|G_T\|+(2+l_{AS})\|G\|$	-	$(2n_S+m_{max}+3)\|G\|+\|G_T\|$	$(2+W)\|G\|$	$(3+2l_{AS})\tau_p+(2l_{AS}+2)G_T$	-
[158]	$\|\xi'\|+2\|G\|$	-	$(2n_S+1)\|G\|$	$(2n_S+1)\|G\|$	$(2n_S+1)\tau_p+(2W+1)G+2n_SG_T$	-
[159]	$3\|G\|$	$2\|G\|$	$(2n_S+5)\|G\|+\|G_T\|$	$4\|G\|$	$3\tau_p+G$	$(3W+1)G$
Ours	$3\|G\|+\|\xi'\|$	$4\|G\|$	$(7+l_{AS}v+n_S)\|G\|+3\|G_T\|$	$(6+\|S+W_{ER}\|)G$	$(4+\|S'+D\|)G+2G_T$	$4\tau_p$

Figure 5.7 illustrates the above relationship. For small n, the percentage of data filtering accuracy drops very fast if $|S'|$ increases. However, when n increases, the accuracy is higher for the same $|S'|$. For example, comparing $n = 10$ and $n = 200$ indicates that for $|S'| = 10$ the accuracy rises from 0 to 95.5% respectively.

The proposed access/threat model is also advantageous in the sense that it decreases the computation and communication burden on the CS in critical situations. Note that the CS first checks that the location of an ER and the time interval within which a data request has occurred (refer to Figure 5.3) are legitimate. Without this prior step, the CS would retrieve, process, and transfer data to the ER.

Table 5.3 compares our scheme with three other works [109, 158, 159] in more detail. In [109], m_{max} is the maximum size of allowed attributes associated with ciphertext. The works [109, 158] are broadcast CP-ABE schemes in which explicit receivers are also specified

within the ciphertext using their identities. The work [159] uses attribute ranges and relations to provide a flexible access policy for CP-ABE. As Table 5.3 indicates, the scheme [109] has better access policy expressiveness in comparison with other works. However, this has an effect on its ciphertext size which is proportional to the number of attributes in the access policy. Other protocols offer constant ciphertext size. In particular, the scheme [158] offers ciphertext size very close to ours. Both of the schemes use a symmetric encryption algorithm to encrypt a message using an element in the target group as the secret key.

The work [159] also offers constant ciphertext size, and has the best secret key size among others. The reason lies in the $(n_{att}, n_{att})-$threshold access structure which brings about a very restrictive access policy. In this case, one can argue that since more storage is easily obtainable and cheap to provide, it is better to offer higher access policy expressiveness and still keep the ciphertext size and decryption computation constant which evidently our scheme offers. Comparing outsourcing communication overhead shows that our scheme is less efficient by a factor of two extra group elements than [159]. However, with such extra elements, our decryption computation overhead on the user side is released from pairing computation. This significantly increases the efficiency of our scheme. To show how big the difference is between group multiplication/exponentiation and pairing computations, as one benchmark, using the BN256 curve with the RELIC library on a modern PC, pairing computation delay is approximately 8.22ms while modular multiplication requires 0.0034ms [167]. This means that the corresponding delay of approximately 2417 group multiplications equals one pairing operation. This was also shown for resource constrained devices in Table 5.2 in which pairing computation for 80-bit security is proportional to approximately 1.27×10^6 group multiplications. Considering the fact that a smartphone lacks powerful resources in comparison with a server, outsourcing such a heavy burden to a more powerful entity increases computation efficiency and decreases decryption delay drastically. In this regard, our scheme outperforms [159]. In terms of computation complexity, the schemes [109] and [158] are inefficient in comparison to ours as a result of the linear relationship between pairing computations and the number of attributes in the access policy/system.

Concerning the delay requirements, the schemes [109] and [158] in comparison with our scheme and [159] have an advantage which can affect computation delay significantly. In this case, the former schemes compute pairings over prime-order groups while the latter ones are based on composite-order groups. Freeman [168] showed that the cost of a Tate pairing computation in composite-order groups on a 1024-bit supersingular curve is 50 times slower than a Tate pairing on a 170-bit MNT curve in prime-order groups. Freeman also showed that pairing computation on a modern PC is done in approximately 150ms on a supersingular curve with $\mathbb{G} \subset E(\mathbb{F}_q) \sim 1024$ bits and $\mathbb{G}_T \subset \mathbb{F}^*_{q^2} \sim 2048$ bits. Using this benchmark, since our outsourced computations require merely 4 pairings, this results in approximately 0.6 seconds of computation delay. Compared to our scheme, the decryption process in [109] and [158] imposes $(3 + 2l_{AS})$ and $(2n_S + 1)$ pairings where l_{AS} and n_S are the number of attributes in the access policy and in the system respectively. A naive comparison shows that those schemes with $l_{AS} = n_S = 100$ impose the same delay as our scheme.

In order to decrease the computation delay, [168, 169] proposed efficient ways to convert composite-order bilinear groups to prime order groups and yet keep the orthogonality property. Freeman [168] proposed to use two groups of the same prime-order (e.g. $\log_2 p = 256$) and an asymmetric bilinear map to provide the orthogonality feature of composite-order (e.g. $\log_2 N = 3072$) groups in prime-order groups. Then, Lewko [169] provided a generic conversion using Dual Pairing Vector Space (DPVS). However, this method has a drawback in which instead of one paring in a composite-order group of n_p primes, it needs $2n_p$ pairings in prime-order groups. Using this conversion for our scheme for which we used a composite-order group comprised of 2 primes, we need 4 pairings in prime-order groups for every single pairing originally. Considering our decryption process which requires 4 pairing calculations, the prime-order conversion of our scheme requires 16 pairings for such a process. Using Freeman's benchmark in which prime-order pairing computation imposes a $3ms$ delay, pairing computation in the decryption process is performed in approximately $50ms$. In comparison with [109] and [158], assuming $l_{AS} = n_S = 8$ attributes for those schemes, the computation delay is similar to ours. However, such system parameters are very limited which highlights the effectiveness of our scheme.

Table 5.4: Comparison of computation delay for prime-order and composite-order groups

Curve/Pairing	$\log_2 n$	**Pairing**	**Exponentiation in $\mathbb{G}_1$**	**Exponentiation in $\mathbb{G}_2$**	**Exponentiation in $\mathbb{G}_T$**
BN256/Ate	256	5.05	0.55	1.91	5.16
Supersingular/Tate	3072	1276.3	556.9	-	174.88

Table 5.4 shows the computation delay comparison between prime-order on a BN256 curve using Ate paring and composite-order with two primes on a supersingular curve using Tate pairing on a 2.6 GHz Intel Celeron 64 bit PC with 1 GB RAM (data extracted from [170]). The table shows the efficiency of such a conversion in terms of computation delay for 128-bit security. The delays are in milliseconds.

5.8 Conclusion

In an emergency, ERs require accurate, timely, and location-aware information. Acquiring such information encounters substantial challenges, among which filtering a large volume of data, privacy, and authorized access have received little attention. To jointly address the aforementioned challenges, this work proposed a location-aware access authorization scheme for emergency response. We integrated BE with CP-ABE to incorporate dynamic attributes (i.e., location and time) into an access policy. The LA-CP-ABE scheme ensures that an authorized ER is able to retrieve relevant, timely, and location-aware information. The performance analysis of LA-CP-ABE indicates the efficiency and effectiveness of the scheme in comparison with state-of-the-art fine-grained authorization schemes. Our scheme imposes constant decryption computation complexity and communication overhead. The use of BE for incorporating the location attribute decreases the communication cost of the updating process. However, there is a trade-off between the updating communication cost and accuracy of data. In this case, the large size of $S'' > 1$ implies a lower updating cost requirement which may lead to lower location-accurate data. In terms of security, the proposed scheme is CCA-selective secure based on the $m-$BDHE assumption and addresses the key escrow problem.

Chapter 6

Trust Solution Framework in Smart Emergency Management

6.1 Introduction

The advent of Social Media Networks (SMNs), the IoT, Smart Grid, and ITS is creating a hyper-connected world where people are always connected and millions of physical things are connected, accessible, and controllable through the Internet [149]. The interaction between these connected entities creates a wealth of data that can be leveraged to realize smart cities, where citizens are engaged, and resources and services are reliably delivered and efficiently used. One of the major beneficiaries of this wealth of data is the public safety sector [3, 34]. By employing smart mobile devices and wireless Internet access, emergency responders can access this wealth of data at any time and in any location. This can help them achieve high levels of situational awareness during an emergency. Nonetheless, this virtue is only achieved when data is accurate, relevant, reliable, and timely. Unfortunately, often times, some of these features might be missing. For instance, existing emergency reporting systems (i.e., 9-1-1 call centres) suffer from high percentages of false emergency reports (i.e., false alarms) that can be as high as 20% of the total call volume [171]. Because an emergency report cannot be easily discredited or ignored, these false alarms end up draining valuable resources. In a hyper-connected world, this will be

more challenging thanks to the countless number of automatic emergency reports (e.g. Smart Grid sensors, Autonomous Vehicles sensors, IoT sensors, etc.).

The preceding problem may even be magnified considering intentionally generated false notifications. Notice that the hyper-connected world offers high connectivity, accessibility, and information availability. This also suggests that intruders and adversaries could also gain access to that enormous interconnected Internet, compromise a number of nodes, and generate intentionally malicious emergency calls/notifications. Generally, a node is an entity in the hyper-connected world such as a person in social media or a sensor in a smart building.

Besides the false notifications, during an emergency, the situation is harsh and dynamic which negatively affects most data sources (e.g., sensors and people), generated data, and transmission of the data. For instance, sensors may malfunction due to high temperature while humans may undergo high levels of stress, which may influence their ability to accurately report their observations [5]. In addition, the threat of malicious activities with regards to data such as data tampering and false data generation is still viable during this time. Therefore, the large volume of data that is generated and shared in the hyper-connected world contains many uncertainties. Moreover, accessing such a volume of data (i.e., raw data) is likely to overwhelm emergency responders.

6.1.1 Actionable Information

In order for emergency responders to fully benefit from the wealth of raw data generated by the hyper-connected world, the raw data should go through various processes and verifications. Such a step is to filter the large volume of data, reducing its uncertainties, and processing it to conform to the requirements of emergency response. The outcome of this step is actionable information used for planning and carrying out emergency operations. This information is accurate, context-aware, reliable, and timely [172]. Automating the extraction of actionable information from a wealth of raw data will give birth to the era of Smart Emergency Response (SER). For example, rumours and speculations regarding an emergency incident should be filtered-out from social media feeds. In addition, the

accuracy of raw sensory data should be verified, and the data should be processed in the context of the ongoing emergency. Data integrity should also be checked as it is transferred through various communication links, some of which may be insecure.

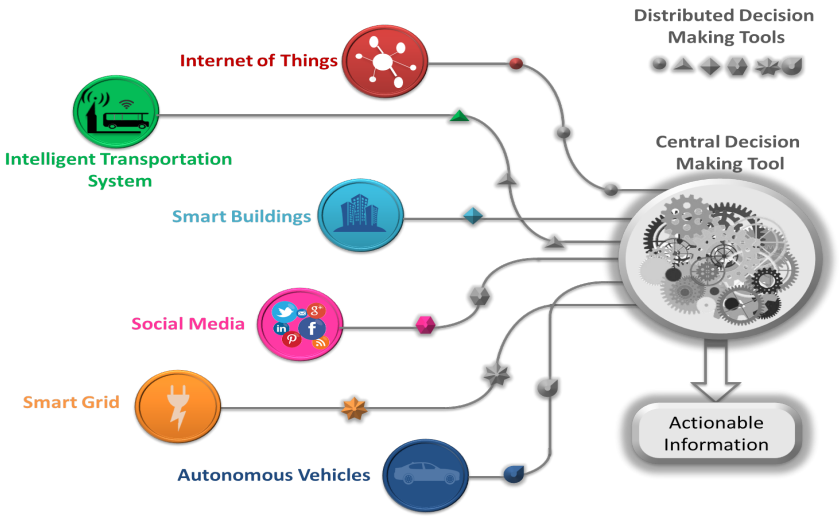

Figure 6.1: A Conceptual model for deriving actionable information in SER

A conceptual model for deriving actionable information is illustrated in Figure 6.1. Here, raw data comes from the elements of the hyper-connected world, traverses the communication links through one or many distributed Decision Making Tools (DMTs), and feeds into a centralized DMT. The distributed DMTs may be comprised of several distinct data processing and filtering units. Examples of such processes are predefined protocols for emergency operations and standard procedures with which an emergency should be handled. The outcome of the model is actionable information. While emergency responders are interested in actionable information, they can also have access to the outcome of the distributed DMTs.

6.1.2 Trust Modeling and Evaluation

Trust modeling and evaluation can be used as a key preliminary filtering tool in DMTs. In SER, trust is defined as a value that reflects the belief or confidence or expectation on the honesty, integrity, ability, availability, and quality of service/information of target nodes future activity/behavior [173]. For example, the opinion of a subject matter expert

regarding an incident is more trustworthy than a random person who states his/her opinion about the same situation.

Employing trust modeling and evaluation has several advantages. First, trust filters large amounts of raw data to a fraction of higher value and quality data. In a smart city where several disjoint sensors may sense the same event, trust can help reduce the number of relayed messages. In addition, at the relaying nodes, trust can be used as a priority measure by which higher-trusted packets can be transmitted first. Second, it reduces the risk factors involved in emergency response operations such as poor situational awareness. Third, trust reduces the cost of operations by detecting false/inaccurate notifications/data, i.e., improving the operational efficiency. Last but not least, establishing trusted relationships among individuals and governmental agencies lowers privacy concerns which may result in increasing data availability. Privacy concern is a significant reason for withholding critical data such as health records. Here, the higher the trust between two entities, the lower becomes the privacy concern.

In the context of SER, trust modeling and evaluation are challenging. Trust is a highly subjective matter [174]. In order to effectively employ trust, the context and dynamics of an emergency, the corresponding influential factors, and the viewpoints of emergency responders in interpreting raw data should be translated into trust modeling. In addition, the heterogeneity of data sources confines the construction of a unified trust computation model. Furthermore, since the criticality of an emergency heightens the significance of delay in data retrieval and processing, trust evaluation on raw data should be done in real-time. Consequently, trust evaluation should not need powerful processing machines. Rather, resource constrained devices (smartphones and tablets) should be adequate. Finally, privacy concerns blind trust evaluations by increasing the inaccuracy ratio within the process. This is because the complete set of required data may not be available for trust evaluations.

In this chapter, we lay the integral foundations for employing the concept of trust in SER. To accomplish that, taking the definition of trust and response into consideration, this chapter unveils the need to develop trust in a broader context of Smart Emergency

Management (SEM). More importantly, we introduce a novel solution framework, which shows where and when trust modeling and evaluation should be employed in SEM. The proposed framework perfectly conforms to the phases and requirements of SEM and identifies its key components that should be taken into consideration in trust modeling and evaluation. The proposed framework is a well-suited starting point for further research in this area. The rest of the chapter is organized as follows. Section III presents a brief background on trust. SEM will be introduced in Section IV and its phases are elaborated. The key enablers of trust modeling and evaluation will be presented in Section V. The proposed framework is presented in Section VI and concluding remarks are made in Section VII.

6.2 Background on Trust

Trust has been studied in various domains including Wireless Sensor Networks (WSNs), Mobile Ad Hoc Networks (MANETs) [173], and SMNs [175]. For every application domain, trust has been defined differently to reflect its implications on that particular domain. For instance, in WSNs trust is defined as a means to evaluate the quality of information. In SMNs, it is used to assess the credibility of the posts regarding an event or a topic. In other applications, it has been used as a subjective probability that an entity conforms to the rules and behaves as expected [176].

Trust computation, propagation, and prediction are called trust dynamics, and they are the foundation of trust modeling and evaluation [176]. In order to evaluate trust on a target node (we call it a subject), a computational model is required. In general, a model consists of several weighted factors extracted from the subjects actions and attributes that affect the trust on that subject. Note that trust is a dynamic phenomenon that changes with time, experience, and context [174, 176]. Trust propagation enables the nodes in a network to access the global or local trust values on other nodes in the network. Predicting the unknown trust value on a node is called trust prediction which can be done using the trend with which the trust value on that node has changed in a period of time.

In SER, trust encompasses all aforementioned domains since data is generated by various sources (sensors and humans) and is communicated through different mediums (WSN, MANET, IoT, etc.). Furthermore, the harsh conditions of an emergency incident may supress the capabilities of data sources to generate high value and high quality information. In addition, such conditions may amplify the criticality of the quality of information and the sternness on the delay requirements corresponding to data availability.

6.3 Trust in Smart Emergency Management

SER is defined as the immediate coordinated effort, using accurate, timely, reliable, and context-aware actionable information, to alleviate the influences of an emergency incident such as loss of life or property [5, 177]. Comparing the definitions of trust and SER indicates that the preliminary steps toward trust modeling and evaluation have to start at a point in time well before an incident occurs (i.e., response is required). In fact, to obtain a more realistic trust value for a subject, one should monitor and study the activity/behaviour of the subject in a period of time before a trust value could be assigned to that subject. To this end, a better way of trust modeling and evaluation for an emergency is by providing a broader concept regarding emergencies to complement SER. This broad concept is Smart Emergency Management (SEM) which is comprised of four consecutive phases in a time continuum. These phases are smart emergency prevention, smart emergency preparedness, SER, and smart emergency recovery. Figure 6.2 illustrates the application of trust in the four phases of SEM.

The first phase, smart emergency prevention, includes structural and non-structural protective measures (e.g., constructing floodways and building fire codes) to reduce the risks of an emergency. In addition, among the elements of the hyper-connected world, for example; IoTs should be deployed, connected, and tested according to the needs of an emergency; buildings should become smart with various sensors. In this phase, potential subjects are identified and trust dynamics are constructed. Moreover, trusted relationships among various sensors, individuals, demographics, public safety personnel and agencies are

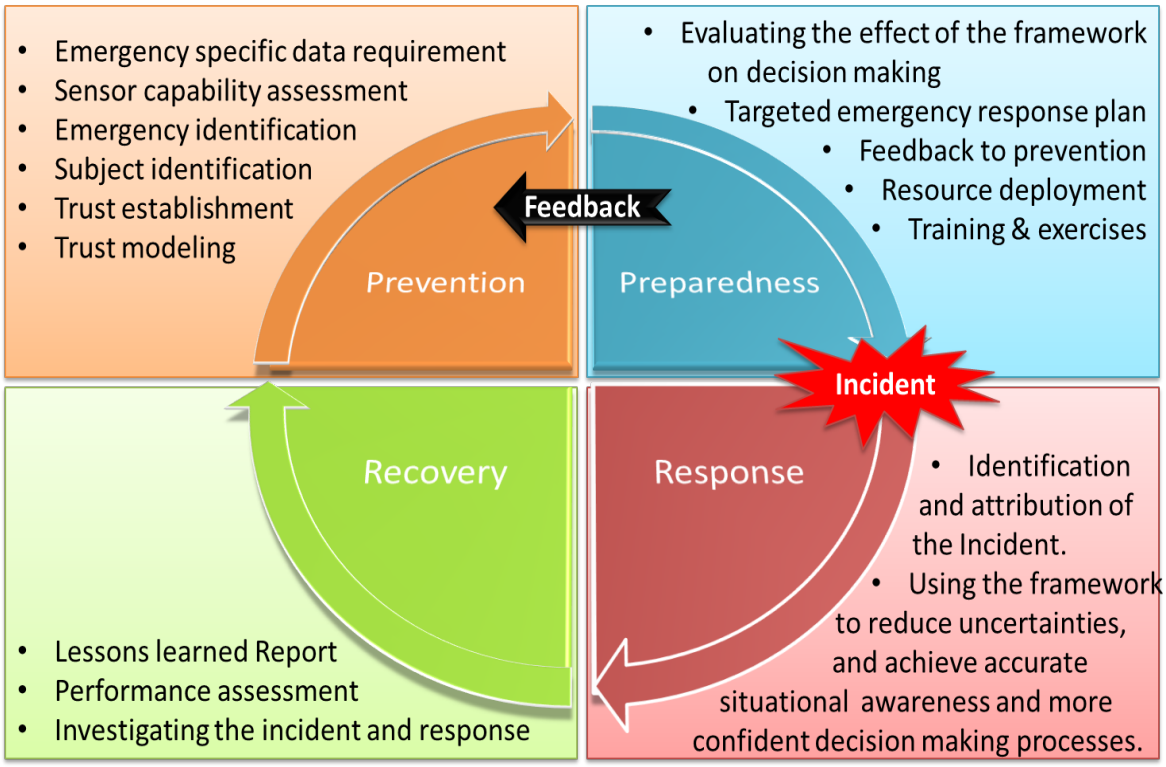

Figure 6.2: Trust in the four phases of SEM

established. Trusted relationships consolidate coordinated operations and increase data availability.

The smart emergency preparedness phase emphasises the operational readiness to respond to an emergency and handle its consequences. Such activities include training and fire drills exercises. For example, the general public should be trained with regards to various emergencies (e.g., the way that they should use social media during an emergency, the keywords they should use, etc.). Through the exercises, trust dynamics and relationships are used and re-evaluated. The activities in this phase provide valuable feedback for the prevention phase in order to augment existing plans, protocols, and procedures. The first two steps recur until an emergency occurs.

At the time of an emergency, the plans and protocols, and the elements of the hyper-connected world will be put in use to immediately and intelligently respond to the emer-

gency. The required data are immediately retrieved and subsequent processes are performed. Here, applying trust evaluation filters-out inaccurate data and uncertainties. Moreover, the dynamics of an emergency are monitored through the trusted elements of the hyper-connected world to increase situational awareness for emergency responders.

Finally, smart emergency recovery includes reconstruction of structures, counseling, and financial assistance, to name a few. In the recovery phase, the advantage and disadvantages of data sources, the retrieved data, and the processing and DMTs are studied. The aftermath reports of an incident will be used to improve the efficiency and effectiveness of emergency response and reduce risks in future incidents. It can be seen that the four phases shape a circle in a respected order such that each phase is affected by its preceding phase and affects its subsequent phase [177].

It is worth stating that the aforementioned four phases need to be taken into consideration separately for every distinct emergency condition. This is because emergencies are of different types, dynamics, and intensity. Different emergencies imply that the respective data requirements may also be different. In addition, the variations of emergencies cause the value of data to change in every emergency. The value of data represents the impact it makes on the decision making process and the respective emergency operation.

6.4 Enablers of Trust Modeling and Evaluation in SEM

In this section, the key enablers of trust modeling and evaluation are introduced. Those are role, subject, and event. These enablers provide a realistic perspective of the structure and the context necessary to construct trust dynamics for SEM. In the following, we will elaborate some details of the enablers and their inter-relations. We also demonstrate how the enablers influence trust modeling and evaluation.

6.4.1 Roles

A target node (i.e., a subject) in SEM can take one or all of the following roles: data generator; data holder; and data consumer. A role declares certain expected actions and

attributes for its possessor, differentiates the impact of those actions and attributes on the trust dynamics for a particular emergency, and in general, defines a context for trust dynamics. For instance, assume that a multi-functional sensor which measures temperature, pressure, and gas density in a room, takes the data generator role in a gas leaking incident. The expected action for this sensor is to generate data representing the density of the gas in its surrounding environment during the incident. In addition, among the sensors attributes, its proximity to the incident and sensing accuracy level gain higher weights in its trust computation model than, e.g., its buffer size. On the other hand, if the sensor takes the data holder role in another incident, the expected action is to store or relay messages from other sensors in the environment. In this case, its buffer size will have higher impact on the trust computation model than its sensing accuracy level.

The aforementioned influences of roles on trust modeling and evaluation assert themselves through four fundamental trust relationships: Consumer-Generator; Consumer-Holder; Holder-Generator; and Consumer-Holder→Generator. The dash shows that there is a bidirectional relationship between two roles and the arrow for the last relationship illustrates that a holder generates new data using the data it has already stored, thus becomes a data generator for which a consumer needs to measure trust accordingly. Any other trust relationship can be narrowed down to these four. These trust relationships are bidirectional, and in each direction, certain factors are to be considered in the trust dynamics.

The Consumer-Generator relationship is formed when a direct contact between, for example, an emergency responder and a patient, occurs in either direction. In this case, having a realistic trust value in both directions profoundly increases the effectiveness of the response and lack of it may have devastating consequences. For example, during the 9/11 incident, a structural engineer advised the incident commander in charge of the emergency operation to evacuate the entire building since it will eventually collapse. However, the incident commander did not pay attention to the advice and evacuation orders were not issued until it was too late.

In the hyper-connected world, considering the significant increase in the number and

accessibility of data generators, the Consumer-Generator relationship is very important. However, constructing and maintaining this relationship before an emergency and evaluating it properly during an emergency are challenging tasks. As the number of data generators is enormous and their diversity is high, constructing a unified computational model of trust is difficult. In addition, keeping a global view of the trust values at every consumer node (e.g., a sensor or a distributed DMT) is highly expensive. It is known that emergencies are local [3]; thus, a localized view point of trust modeling and evaluating is a proper starting point.

The Consumer-Holder relationship happens when, for instance, a data retrieval request is sent to a server or data is relayed through one/many forwarding node(s). In this case, no additional data processing is done when data is stored/relayed in/by a holder. Note that a holder can take many shapes; for example, a mobile phone, a cloud server, or a human observer. Any shape carries a set of attributes for a holder that should be differentiated in trust modeling and evaluation. This relationship can be extended to Consumer-Holder-Generator, if trust in the data generator is also of concern. Observe that the holder receives data from a generator and stores it. In this case, the end-to-end trust evaluation should consider all the entities between a data generator and a consumer inclusive. The increase in the length of the chain here magnifies the difficulty of trust evaluation. In this respect, it would be appealing to have distributed DMTs to evaluate trust en-route.

The Holder-Generator relationship is formed when a data generator transfers data to a database (either centralized or distributed). As with the previous relationship, the holder here does not process data. This relationship imposes the least challenges in trust modeling and evaluation since data transfer of this type occurs mainly before an emergency. Finally, Consumer-Holder→Generator occurs when a consumer sends a data request to a holder; however, the holder performs certain processes on the data to generate new data. For example, in high impact incidents like the Haiti earthquake, several groups of digital volunteers were formed. Those were experienced individuals who were familiar with the emergency/disaster response procedures and the incidents surrounding areas. During that incident, they monitored social media and the dedicated websites for the incident to collect

and process data. Then, they sent the newly generated data to respective authorities for further processing.

6.4.2 Subjects

We define a subject as an entity that plays a role in an event. Subjects include emergency responders, the general public, sensors, community leaders, physical and digital volunteers, autonomous vehicles, smart meters, and so forth. A subject possesses a set of attributes and can participate in various events. Considering trust evaluation of a subject, we should first assign a role to the subject. A role corresponds to several expected actions and attributes. Note that a complete set of actions and attributes of a subject may be beyond the role that is taken by the subject. Therefore, at the time of trust modeling and evaluation for a specific emergency situation, those actions and attributes may outweigh others as mentioned before. Since emergencies may create harsh environments, capability assessments on various subjects should be done to learn the operational boundaries in which the subject is to be trusted.

6.4.3 Events

Before, during, and after any typical emergency scenario, several events may take place: data sensing; data relaying; data processing; and data consuming. Depending on the scenario, data collection and aggregation may also be included. In addition, data processing may have other sub-events such as data retrieving. Note that considering the relaying event, the attributes of a communication link are likely to have an effect on the trust dynamics. The events are useful assets for problem segmentation, recognizing all of the subjects and roles involved, and formulating a comprehensive trust modeling and evaluation. Having particular data at hand is an indicator of the events necessary for the data to be generated and delivered to a consumer. Therefore, events complement roles as they can be used to form a complete chain of trust relationships.

6.5 A Trust Solution Framework for SEM

In this section, we propose a novel solution framework that illustrates when and where the trust dynamics should be constructed and used in the aforementioned four phases of SEM. In addition, the barriers to accomplish the outcomes of the trust modeling and evaluation are identified. The proposed framework appears to be a well-structured projection of the way that the large volume of data and its uncertainties can be filtered-out using trust. Before we dive into the demonstration of the framework, and since data is in the center of SEM, we will first elaborate the different types of data that are available in SEM.

6.5.1 Types of Data in an Emergency

There are two types of data in an emergency; dynamic and static. Dynamic data is changing as the emergency progresses. Therefore, continuous sampling of dynamic data is required in order to keep the situational awareness up-to-date. Examples of dynamic data are any sensory data such as temperature, smoke, pressure, occupants' location, heart rate of emergency responders, skin temperature of firefighters, and posts on twitter regarding the progress of an emergency. Trust in such data plays a crucial role in emergency operations because dynamic data highlights the dynamics of the emergency and the pace with which it progresses. Here, trust modeling and evaluation should reflect the effects of such dynamics. For example, as sensors reach their operational boundaries, the trust in their generated data should be lowered. On the other hand, static data does not change (or hardly changes) as the emergency situation continues. Instances of static data are a building's floor plans, location of the Knox-box, and location of emergency exits.

6.5.2 Trust Solution Framework for Data Filtering in SEM

The proposed framework is illustrated in Figure 6.3. The framework is emergency specific in all four phases which means that for a particular emergency scenario, trust model and evaluation, the value of data and the data sources, the roles, and subjects may vary.

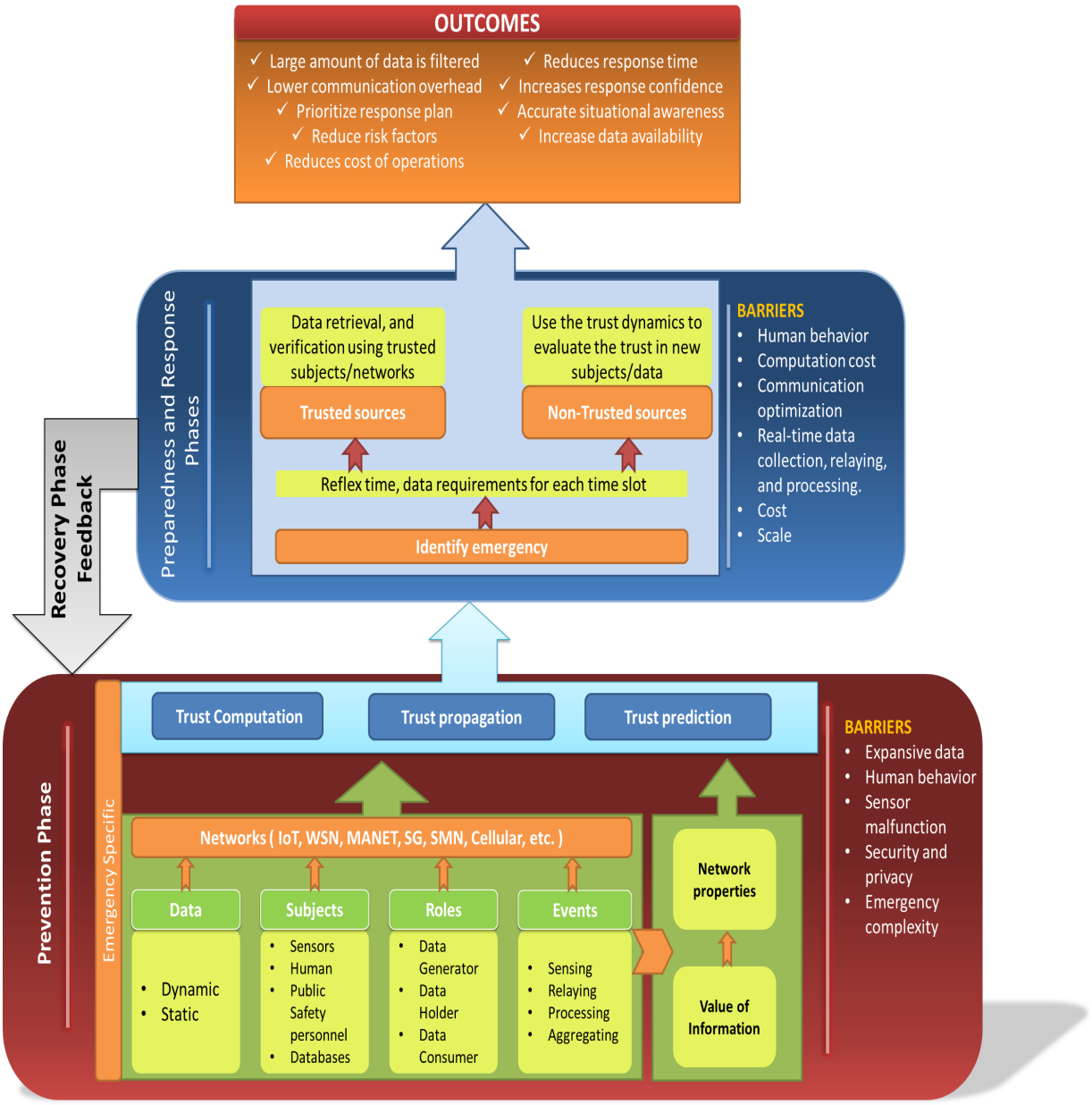

Figure 6.3: Trust modeling and evaluation framework for SEM

In every emergency, there is a certain reflex time continuum in which from the start of an emergency to its mitigation, several distinct time slots are distinguished. Each of the time slots corresponds to a specific set of tasks and information requirements. Putting trust modeling and evaluation in this context will improve the efficiency and effectiveness of SEM.

The prevention phase starts with identifying an emergency, the corresponding reflex

time continuum, and respective information requirements. Then, the enablers of trust dynamics can be identified. To propose trust computation, propagation, and prediction models, the data types and value of information, the subjects involved and their roles, the respective trust relationships, events, and the communication infrastructure through which data is transferred need to be taken into consideration. Here, the main barriers include expansive data, human behaviour, security and privacy, and emergency complexity.

The preparedness phase will be launched after the prevention phase in the form of exercises and training to evaluate the accuracy of trust dynamics and their capability to fulfil their goals for a particular presumed emergency. Such studies will be provided as feedback to the prevention phase in order to augment the existing trust dynamics. Then, in an emergency, trust dynamics will be used to filter the large volume of data and its uncertainties. Here, the trusted resources are identified from which the required data will be retrieved. Besides, in the case of other resources for which no trust modeling is available, the existing trust dynamics that closely model and evaluate them can be used. The barriers of these two phases include human behavior, computation cost, communication optimization, real-time data collection, relaying, and processing, and operational costs and scales. After an emergency, the aftermath reports of the incident will provide valuable feedback to improve the framework. Among the main improvements is the identification of the new sources of data for which trust modeling has not been proposed.

For example, for a fire incident, reflex time includes dispatch, turnout, response, access time, setup, and response plan execution. The dispatch time starts from ignition, and includes detection and reporting. This time interval continues until the dispatcher learns about the incident, its causes, location, and determines the required resources and notifies the response units. Very important data is retrieved and processed during dispatch time, and in fact, this is the time that false notifications should be identified. Here, the trusted subjects that have been identified in the prevention and preparedness phases could be queried to verify the authenticity and legitimacy of the incident. The turnout time is the amount of time emergency responders take from the notification to the point of response. The amount of time the responders are en-route until they arrive at the scene is called

response time. During the response time, the emergency responders require information regarding, e.g., the surroundings of the emergency location, the emergency exit doors or routes, floor plans of the building, or map of the area.

Once responders arrive at the scene and the apparatus stops, the access time starts in which the responders move to the exact location of the emergency. From this point in time, they require another set of important information among which are the location of endangered individuals, their health condition, and sensory data. Here, trust evaluation is required to filter the uncertainties, prioritize the response plan, reduce risk factors and cost of the operation, reduce response time, and achieve accurate situational awareness. Setup time is the amount of time required to connect hose lines to fire hydrants and position ladders. Finally, a response plan is executed.

6.6 Conclusion

SEM is going to be realized using accurate, timely, reliable, and context-aware actionable information generated by the elements of the hyper-connected world such as IoT. However, the path towards such a realization is challenging as a large volume of data is generated by such elements. In addition, many uncertainties are entwined with data or produced by malicious entities. In this chapter, the concept of trust is introduced as an effective tool to address the aforementioned challenges. Trust increases situational awareness for emergency responders and improves the effectiveness of emergency operations. We proposed a novel framework that illustrates when, where, and how trust dynamics should be constructed and used in SEM. In addition, the framework identifies the barriers to constructing and applying trust dynamics in SEM and shows the open areas of research.

Chapter 7

Conclusion and Future Directions

7.1 Concluding Remarks

PSBNs are going to facilitate interoperability among ERs and provide the means to increase data availability during an emergency. Unfortunately, in the commercial communications technologies such as LTE (the leading enabler technology of PSBNs), several security and privacy vulnerabilities have been detected. Considering the criticality of an emergency and the strict security and privacy requirements of PSBNs, such vulnerabilities should be mitigated. On the other hand, in an emergency, accurate, timely, and context and location-aware information is required. Acquiring such information encounters substantial challenges among which data availability, filtering a large volume of data, privacy, and authorized access have received little attention.

To address the aforementioned challenges, the contributions of this thesis are four fold. First, an efficient algorithm was proposed to address the main vulnerability of the LTE which is the permanent subscriber's identity threat. This algorithm uses AES symmetric encryption along with a secure shuffling technique to permute the preloaded shared keys. The proposed algorithm was heuristically shown to be secure and comparative performance evaluation showed its superiority over similar algorithms and well-known public-key encryption schemes in terms of communication and computation overheads and delay.

Second, a secure data storage structure along with a privacy-preserving search algorithm was proposed to provide context and location-aware data availability. Security analysis showed that the scheme is secure against a chosen keyword attack and privacy of DOs and ERs is preserved. Our search process imposes $O(1)$ delay which is ideal for emergency situations. Moreover, the scheme offers conjunctive and disjunctive keyword search which enables the ERs to search for more than one word at a time. This decreases the overall communication overhead for most cases. Furthermore, the proposed scheme supports dynamic movements of DOs in the sense that it enables light-weight updating of the data entry. In addition, in comparison with the state of the art algorithms the scheme has better computational complexity and delay.

Third, a location-aware access authorization scheme was proposed to filter out irrelevant data and provide privacy for DOs and ERs. We integrated BE with CP-ABE to incorporate dynamic attributes (i.e., location and time) into an access policy. The proposed scheme ensures relevant, timely, and location-aware information access authorization. The performance analysis of the proposed scheme indicates the efficiency and effectiveness of the scheme in comparison with state-of-the-art fine-grained authorization schemes. In this case, the scheme imposes constant decryption computation complexity and communication overhead. The cost of updating ciphertext is decreased by employing BE. However, there is a trade-off between the updating communication cost and accuracy of data. In this case, the large size of $S'' > 1$ implies a lower updating cost requirement which may lead to lower location-accurate data. In terms of security, the proposed scheme is CCA-selective secure based on the $m-$BDHE assumption and addresses the key escrow problem.

Fourth, we proposed a framework that facilitates realizing SEM by filtering out a large volume of data. Since many uncertainties are entwined with data or produced by malicious entities, the framework uses the concept of trust as an effective tool to filter out the large volume of data. Trust increases SA for ERs and improves the effectiveness of emergency operations. Our novel framework illustrates when, where, and how trust dynamics should be constructed and used in SEM. In addition, the framework identifies the barriers to constructing and applying trust dynamics in SEM and shows the open areas of research.

Overall, availability of actionable information is critical for effective emergency response. Actionable information is characterized by high accuracy, context and location awareness, reliability, and timeliness. In addition, using actionable information should comply with privacy rules and access authorization policies. In this thesis, we proposed several innovative schemes to provide actionable information availability while respecting privacy and data access authorization requirements.

7.2 Future Directions

Based on the investigations and findings in this thesis, there are several areas that can be considered for future research. These areas can be summarized as follows:

- 5G is going to be the future of communications technology in which everything and everybody is connected and accessible at any time and location. In this regard, the effect of 5G on public safety communications should be investigated. In addition, the strict requirements of PSBNs such as security and privacy should be respected.

- Lightweight data availability and access authorization schemes are required with respect to 5G specifications and requirements.

- Data availability is fundamental in effective emergency response. In our scheme, we used mobile cloud entities to facilitate data access for ERs. Mobile clouds may only be able to support a limited number of DOs. Thus, this work can be extended by a novel dynamic and opportunistic memory management scheme to accommodate a large number of DOs in a certain location area.

- Data access authorization schemes should enable DOs to define their preferred access policy. Although our scheme provided such a key requirement, an open area of research is to propose a novel access authorization scheme that provides constant computation and communication overheads yet the access policy supports both conjunctive and disjunctive attributes. Our authorization scheme lacks supporting disjunctive attributes.

- The implementation of the proposed algorithms in real test-beds and investigating the effects of those algorithms on data availability and achieving high levels of situational awareness are appealing.

- Our data filtering framework based on trust opens unique and extremely important and influential areas of research. For example, using the framework, concrete data filtering schemes are required for various emergency scenarios. This is because every emergency scenario is different from others and has varied data requirements. This implies different data sources with different characteristics.

References

[1] T. Doumi, M. Dolan, S. Tatesh, A. Casati, G. Tsirtsis, K. Anchan, and D. Flore, "LTE for public safety networks," *Communications Magazine, IEEE*, vol. 51, no. 2, pp. 106–112, February 2013.

[2] Harvard Humanitarian Initiative, "Disaster relief 2.0: The future of information sharing in humanitarian emergencies," UN Foundation and Vodafone Foundation Technology Partnership, 2011.

[3] "Research roadmap for smart fire fighting," National Institute of Standards and Technology (NIST), 2015. [Online]. Available: www.nfpa.org/SmartFireFighting

[4] J. P. Blair and K. W. Schweit, "A study of active shooter incidents, 2000 - 2013," Texas State University and Federal Bureau of Investigation, U.S. Department of Justice, Washington D.C., 2014.

[5] Edited by Jeffrey A. Larsen, "Responding to catastrophic events," Palgrave Macmillan, 2013.

[6] A. Bikos and N. Sklavos, "LTE/SAE security issues on 4G wireless networks," *IEEE Security Privacy*, vol. 11, no. 2, pp. 55–62, March 2013.

[7] N. Seddigh, B. Nandy, R. Makkar, and J. F. Beaumont, "Security advances and challenges in 4G wireless networks," in *Eighth Annual International Conference on Privacy Security and Trust (PST)*, Ottawa, Canada, Aug 2010, pp. 62–71.

[8] J. L. Barr, E. R. Burtner, W. Pike, A. M. B. Peddicord, and B. Minsk, "Gap assessment in the emergency response community," Homeland Security (US), Report No.: PNNL-19782, September 2010.

[9] J. L. Barr, A. M. B. Peddicord, E. R. Burtner, and H. A. Mahy, "Current domain challenges in the emergency response community," in *Proceedings of the 8th International ISCRAM Conference–Lisbon*, vol. 1, 2011.

[10] "Public safety homeland security bureau, federal communications commission", "the public safety broadband wireless network: 21st century communications for first responders," Mar. 2010.

[11] R. Ferrús, O. Sallent, G. Baldini, and L. Goratti, "LTE: The technology driver for future public safety communications," *IEEE Communications Magazine*, vol. 51, no. 10, pp. 154–161, 2013.

[12] "Long term evolution (LTE): A technical overview." Motorola, 2007, Tech. Rep., 2007. [Online]. Available: http://www.motorola.com

[13] M. Simic, "Feasibility of long term evolution (LTE) as technology for public safety," in *20th IEEE Telecommunications Forum (TELFOR)*, Belgrade, Serbia, Nov 2012, pp. 158–161.

[14] J. Harrald and T. Jefferson, "Shared situational awareness in emergency management mitigation and response," in *40th Annual Hawaii International Conference on System Sciences. HICSS 2007.*, Waikoloa, HI, Jan 2007, pp. 23–23.

[15] I. Bilogrevic, M. Jadliwala, and J.-P. Hubaux, "Security issues in next generation mobile networks: LTE and femtocells," in *2nd International Femtocell Workshop, Luton, UK*, no. EPFL-POSTER-149153, Luton, UK, 2010.

[16] D. Caragata, S. El Assad, C. Shoniregun, and G. Akmayeva, "UMTS security: Enhancement of identification, authentication and key agreement protocols," in *International Conference for Internet Technology and Secured Transactions (ICITST)*, Abu Dhabi, UAE, Dec 2011, pp. 278–282.

[17] H. Choudhury, B. Roychoudhury, and D. Saikia, "Enhancing user identity privacy in LTE," in *IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, Liverpool, England, June 2012, pp. 949–957.

[18] D. Yu and W. Wen, "Non-access-stratum request attack in E-UTRAN," in *Computing, Communications and Applications Conference (ComComAp)*, Hong Kong, Jan 2012, pp. 48–53.

[19] B. Ravishankar and M. Harishankar, "Roaming issues in 3gpp security architecture and solution using umm architecture," in *The Second International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies. UBICOMM '08.*, Valencia, Spain, Sept 2008, pp. 457–462.

[20] C. Anderson, P. Breimyer, S. Foster, K. Geyer, J. Daniel Griffith, A. Heier, A. Majumdar, D. Shah, O. Simek, N. Stanisha, and F. Waugh, "A network science approach to open source data fusion and analytics for disaster response," in *18th International Conference on Information Fusion (Fusion)*, July 2015, pp. 207–214.

[21] "Google person finder," Last Visited, November 2016. [Online]. Available: https://google.org/personfinder/global/home.html

[22] "Disaster response on facebook," Last Visited, November 2016. [Online]. Available: https://www.facebook.com/disaster/

[23] "Microsoft solutions for good." Last Visited, November 2016. [Online]. Available: http://www.microsoft.com/about/corporatecitizenship/en-us/nonprofits/solutions-for-good/

[24] S. R. Veil, T. Buehner, and M. J. Palenchar, "A work-in-process literature review: Incorporating social media in risk and crisis communication," *Journal of Contingencies and Crisis Management*, vol. 19, no. 2, pp. 110–122, 2011. [Online]. Available: http://dx.doi.org/10.1111/j.1468-5973.2011.00639.x

[25] K. Kaminska, P. Dawe, K. Forbes, D. Duncan, I. Becking, B. Rutten, and D. ODonnell, "Digital volunteer supported recovery operations experiment," *Defence Research and Development Canada–Scientific Report, DRDC-RDDC-2015-R035*, 2015.

[26] K. Starbird and L. Palen, ""voluntweeters": Self-organizing by digital volunteers in times of crisis," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '11. New York, NY, USA: ACM, 2011, pp. 1071–1080. [Online]. Available: http://doi.acm.org/10.1145/1978942.1979102

[27] A. Gupta and P. Kumaraguru, "Credibility ranking of tweets during high impact events," in *Proceedings of the 1st Workshop on Privacy and Security in Online Social Media*, ser. PSOSM '12. New York, NY, USA: ACM, 2012, pp. 2:2–2:8. [Online]. Available: http://doi.acm.org/10.1145/2185354.2185356

[28] A. L. Hughes and L. Palen, "Twitter adoption and use in mass convergence and emergency events," *International Journal of Emergency Management*, vol. 6, no. 3-4, pp. 248–260, 2009.

[29] A. Acar and Y. Muraki, "Twitter for crisis communication: lessons learned from japan's tsunami disaster," *International Journal of Web Based Communities*, vol. 7, no. 3, pp. 392–402, 2011.

[30] M. Abu-Elkheir, H. S. Hassanein, and S. M. A. Oteafy, "Enhancing emergency response systems through leveraging crowdsensing and heterogeneous data," in *2016 International Wireless Communications and Mobile Computing Conference (IWCMC)*, Sept 2016, pp. 188–193.

[31] "Smart911," Last Visited, November 2016. [Online]. Available: https://www.smart911.com

[32] J. Li, Q. Li, C. Liu, S. U. Khan, and N. Ghani, "Community-based collaborative information system for emergency management," *Computers & Operations Research*, vol. 42, pp. 116 – 124, 2014.

[33] B. Carminati, E. Ferrari, and M. Guglielmi, "A system for timely and controlled information sharing in emergency situations," *IEEE Transactions on Dependable and Secure Computing*, vol. 10, no. 3, pp. 129–142, May 2013.

[34] D. Boyd, "Public safety statement of requirement for communication and interoperability," U.S. Department of Homeland Security (DHS), The SAFECOM program, 2006.

[35] "The next generation of communication networks and services, the 5G infrastructure public private partnership (5GPPP)," European Commission, 2015. [Online]. Available: http://5g-ppp.eu/wp-content/uploads/2015/02/5G-Vision-Brochure-v1.pdf

[36] M. Agiwal, A. Roy, and N. Saxena, "Next generation 5G wireless networks: A comprehensive survey," *IEEE Communications Surveys Tutorials*, vol. 18, no. 3, pp. 1617–1655, thirdquarter 2016.

[37] N. Panwar, S. Sharma, and A. K. Singh, "A survey on 5G: The next generation of mobile communication," *Physical Communication*, vol. 18, Part 2, pp. 64 – 84, 2016, special Issue on Radio Access Network Architectures and Resource Management for 5G.

[38] Z. S. Bojkovic, M. R. Bakmaz, and B. M. Bakmaz, "Research challenges for 5G cellular architecture," in *12th International Conference on Telecommunication in Modern Satellite, Cable and Broadcasting Services (TELSIKS)*, Oct 2015, pp. 215–222.

[39] S. Borkar and H. Pande, "Application of 5G next generation network to internet of things," in *2016 International Conference on Internet of Things and Applications (IOTA)*, Jan 2016, pp. 443–447.

[40] S. Rajendran, B. V. den Bergh, T. Vermeulen, and S. Pollin, "Ieee 5G spectrum sharing challenge: A practical evaluation of learning and feedback," *IEEE Communications Magazine*, vol. 54, no. 11, pp. 210–216, November 2016.

[41] R. H. Tehrani, S. Vahid, D. Triantafyllopoulou, H. Lee, and K. Moessner, "Licensed spectrum sharing schemes for mobile operators: A survey and outlook," *IEEE Communications Surveys Tutorials*, vol. 18, no. 4, pp. 2591–2623, Fourthquarter 2016.

[42] W. Xiang, K. Zheng, and X. S. Shen, "5G mobile communications," Springer, 2016.

[43] S. Andreev, O. Galinina, A. Pyattaev, J. Hosek, P. Masek, H. Yanikomeroglu, and Y. Koucheryavy, "Exploring synergy between communications, caching, and computing in 5G-grade deployments," *IEEE Communications Magazine*, vol. 54, no. 8, pp. 60–69, August 2016.

[44] C. Yu, W. Hou, Y. Guan, Y. Zong, and P. Guo, "Virtual 5G network embedding in a heterogeneous and multi-domain network infrastructure," *China Communications*, vol. 13, no. 10, pp. 29–43, Oct 2016.

[45] M. H. Eiza, Q. Ni, and Q. Shi, "Secure and privacy-aware cloud-assisted video reporting service in 5G-enabled vehicular networks," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 10, pp. 7868–7881, Oct 2016.

[46] A. Kumbhar, F. Koohifar, I. Guvenc, and B. Mueller, "A survey on legacy and emerging technologies for public safety communications," *IEEE Communications Surveys Tutorials*, vol. PP, no. 99, pp. 1–1, 2016.

[47] "Privacy act", r.s.c. 1985, c. p-21, "http://laws-lois.justice.gc.ca/eng/acts/p-21/," Last Visited, November 2016.

[48] "Personal information protection and electronic documents act", S.C. 2000, c. 5, "http://laws-lois.justice.gc.ca/eng/acts/p-8.6/," Last Visited, November 2016.

[49] Wikipedia, "Personally identifiable information", "http://en.wikipedia.org/wiki/personally identifying information," Last Visited, November 2016.

[50] J. Cao, M. Ma, H. Li, Y. Zhang, and Z. Luo, "A survey on security aspects for lte and lte-a networks," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 283–302, 2014.

[51] 3GPP, "3GPP system architecture evolution (SAE); security architecture," in *TS 33.401, V9.7.0*, 2011.

[52] J. Abdo, H. Chaouchi, and M. Aoude, "Ensured confidentiality authentication and key agreement protocol for EPS," in *Symposium on Broadband Networks and Fast Internet (RELABIRA)*, Baabda, Lebanon, May 2012, pp. 73–77.

[53] L. Xiehua and W. Yongjun, "Security enhanced authentication and key agreement protocol for LTE/SAE network," in *7th IEEE International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM)*, Wuhan, China, Sept 2011, pp. 1–4.

[54] Y. Zheng, D. He, X. Tang, and H. Wang, "AKA and authorization scheme for 4G mobile networks based on trusted mobile platform," in *5th IEEE International Conference on Information Communications & Signal Processing*, 2005, pp. 976–980.

[55] D. He, J. Wang, and Y. Zheng, "User authentication scheme based on self-certified public-key for next generation wireless network," in *IEEE International Symposium on Biometrics and Security Technologies.*, 2008, pp. 1–8.

[56] H. Mun, K. Han, and K. Kim, "3G-WLAN interworking: security analysis and new authentication and key agreement based on EAP-AKA," in *IEEE Wireless Telecommunications Symposium, 2009. WTS 2009*, 2009, pp. 1–8.

[57] J. V. Franklin and K. Paramasivam, "Enhanced authentication protocol for improving security in 3GPP LTE networks," in *International Conference on Information and Network Technology*, 2011, pp. 28–33.

[58] C.-E. Vintilă, V.-V. Patriciu, and I. Bica, "Security analysis of LTE access network," in *Proceeding of 10th International Conference on Networks*, 2011, pp. 29–34.

[59] D. Forsberg, H. Leping, K. Tsuyoshi, and S. Alanara, "Enhancing security and privacy in 3GPP E-UTRAN radio interface," in *IEEE 18th International Symposium on Personal, Indoor and Mobile Radio Communications. PIMRC 2007.*, Athens, Greece, Sept 2007, pp. 1–5.

[60] T. Ta and J. S. Baras, "Enhancing privacy in lte paging system using physical layer identification," in *Data Privacy Management and Autonomous Spontaneous Security*. Pisa, Italy: Springer, 2013, pp. 15–28.

[61] A. R. Beresford and F. Stajano, "Location privacy in pervasive computing," *IEEE Pervasive computing*, vol. 2, no. 1, pp. 46–55, 2003.

[62] M. Li, S. Yu, N. Cao, and W. Lou, "Authorized private keyword search over encrypted data in cloud computing," in *31st IEEE International Conference on Distributed Computing Systems (ICDCS)*, 2011, pp. 383–392.

[63] B. K. Samanthula, Y. Elmehdwi, G. Howser, and S. Madria, "A secure data sharing and query processing framework via federation of cloud computing," *Information Systems*, vol. 48, pp. 196 – 212, Elsevier, 2015.

[64] J. Sun, X. Zhu, C. Zhang, and Y. Fang, "HCCP: Cryptography based secure ehr system for patient privacy and emergency healthcare," in *31st International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2011, pp. 373–382.

[65] D. X. Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," in *IEEE Symposium on Security and Privacy, 2000. S P 2000.*, Berkeley, CA,USA, 2000, pp. 44–55.

[66] Y.-C. Chang and M. Mitzenmacher, "Privacy preserving keyword searches on remote encrypted data," in *Applied Cryptography and Network Security*. Springer, 2005, pp. 442–455.

[67] E.-J. Goh *et al.*, "Secure indexes." *IACR Cryptology ePrint Archive*, vol. 2003, pp. 216–234, 2003.

[68] C. Dong, G. Russello, and N. Dulay, "Shared and searchable encrypted data for untrusted servers," *Journal of Computer Security*, vol. 19, no. 3, pp. 367–397, 2011.

[69] M. Raykova, B. Vo, S. M. Bellovin, and T. Malkin, "Secure anonymous database search," in *Proceedings of the 2009 ACM Workshop on Cloud Computing Security*,

ser. CCSW '09. Chicago, Illinois, USA: ACM, 2009, pp. 115–126. [Online]. Available: http://doi.acm.org/10.1145/1655008.1655025

[70] M. Bellare, A. Boldyreva, and A. ONeill, "Deterministic and efficiently searchable encryption," in *Advances in Cryptology–CRYPTO 2007*. Springer, 2007, pp. 535–552.

[71] L. Ballard, S. Kamara, and F. Monrose, "Achieving efficient conjunctive keyword searches over encrypted data," in *Information and Communications Security*. Springer, 2005, pp. 414–426.

[72] J. Li, Q. Wang, C. Wang, N. Cao, K. Ren, and W. Lou, "Fuzzy keyword search over encrypted data in cloud computing," in *IEEE INFOCOM*, San Diego, CA, USA, March 2010, pp. 1–5.

[73] V. Levenstein, "Binary codes capable of correcting spurious insertions and deletions of ones," *Problems of Information Transmission*, vol. 1, no. 1, pp. 8–17, 1965.

[74] B. Hore, S. Mehrotra, M. Canim, and M. Kantarcioglu, "Secure multidimensional range queries over outsourced data," *The International Journal on Very Large Data Bases (VLDB)*, vol. 21, no. 3, pp. 333–358, 2012.

[75] D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search," in *Advances in Cryptology-Eurocrypt 2004*. Springer, 2004, pp. 506–522.

[76] J. Baek, R. Safavi-Naini, and W. Susilo, "Public key encryption with keyword search revisited," in *Computational Science and Its Applications–ICCSA 2008*. Springer, 2008, pp. 1249–1259.

[77] M. Abdalla, M. Bellare, D. Catalano, E. Kiltz, T. Kohno, T. Lange, J. Malone-Lee, G. Neven, P. Paillier, and H. Shi, "Searchable encryption revisited: Consistency properties, relation to anonymous IBE, and extensions," in *Advances in Cryptology–CRYPTO 2005*. Springer, 2005, pp. 205–222.

[78] Q. Liu, G. Wang, and J. Wu, "Secure and privacy preserving keyword searching for cloud storage services," *Journal of network and computer applications*, vol. 35, no. 3, pp. 927–933, Elsevier, 2012.

[79] R. Curtmola, J. Garay, S. Kamara, and R. Ostrovsky, "Searchable symmetric encryption: Improved definitions and efficient constructions," in *Proceedings of the 13th ACM Conference on Computer and Communications Security*, ser. CCS '06. Alexandria, Virginia, USA: ACM, 2006, pp. 79–88. [Online]. Available: http://doi.acm.org/10.1145/1180405.1180417

[80] Y. Tong, J. Sun, S. Chow, and P. Li, "Cloud-assisted mobile-access of health data with privacy and auditability," *IEEE Journal of Biomedical and Health Informatics*, vol. 18, no. 2, pp. 419–429, March 2014.

[81] K. Kurosawa and Y. Ohtaki, "UC-secure searchable symmetric encryption," in *Financial Cryptography and Data Security*. Springer, 2012, pp. 285–298.

[82] D. Cash, S. Jarecki, C. Jutla, H. Krawczyk, M.-C. Roşu, and M. Steiner, "Highly-scalable searchable symmetric encryption with support for boolean queries," in *Advances in Cryptology–CRYPTO 2013*. Springer, 2013, pp. 353–373.

[83] M. Chase and S. Kamara, "Structured encryption and controlled disclosure," in *Advances in Cryptology–ASIACRYPT 2010*. Springer, 2010, pp. 577–594.

[84] D. Cash, J. Jaeger, S. Jarecki, C. Jutla, H. Krawczyk, M.-C. Rosu, and M. Steiner, "Dynamic searchable encryption in very large databases: Data structures and implementation," in *21st Annual Network and Distributed System Security Symposium, NDSS*, 2014.

[85] S. Jarecki, C. Jutla, H. Krawczyk, M. Rosu, and M. Steiner, "Outsourced symmetric private information retrieval," in *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, ser. CCS '13. New York, NY, USA: ACM, 2013, pp. 875–888. [Online]. Available: http://doi.acm.org/10.1145/2508859.2516730

[86] N. Cao, C. Wang, M. Li, K. Ren, and W. Lou, "Privacy-preserving multi-keyword ranked search over encrypted cloud data," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 1, pp. 222–233, 2014.

[87] M. J. Freedman, Y. Ishai, B. Pinkas, and O. Reingold, "Keyword search and oblivious pseudorandom functions," in *Theory of Cryptography*. Springer, 2005, pp. 303–324.

[88] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Advances in cryptology, EUROCRYPT99*. Springer, 1999, pp. 223–238.

[89] P. Golle, J. Staddon, and B. Waters, "Secure conjunctive keyword search over encrypted data," in *Applied Cryptography and Network Security*. Springer, 2004, pp. 31–45.

[90] D. Boneh, C. Gentry, S. Halevi, F. Wang, and D. J. Wu, "Private database queries using somewhat homomorphic encryption," in *Applied Cryptography and Network Security*. Springer, 2013, pp. 102–118.

[91] Z. Brakerski, "Fully homomorphic encryption without modulus switching from classical GapSVP," in *Advances in Cryptology–CRYPTO 2012*. Springer, 2012, pp. 868–886.

[92] Z. Brakerski, C. Gentry, and V. Vaikuntanathan, "(leveled) fully homomorphic encryption without bootstrapping," in *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, ser. ITCS '12. Cambridge, Massachusetts, USA: ACM, 2012, pp. 309–325. [Online]. Available: http://doi.acm.org/10.1145/2090236.2090262

[93] E. De Cristofaro, Y. Lu, and G. Tsudik, "Efficient techniques for privacy-preserving sharing of sensitive information," in *Trust and Trustworthy Computing*. Springer, 2011, pp. 239–253.

[94] M. Barua, X. Liang, R. Lu, and X. Shen, "PEACE: An efficient and secure patient-centric access control scheme for ehealth care system," in *IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2011, pp. 970–975.

[95] J. Zhou, X. Lin, X. Dong, and Z. Cao, "PSMPA: Patient self-controllable and multi-level privacy-preserving cooperative authentication in distributedm-healthcare cloud computing system," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 6, pp. 1693–1703, June 2015.

[96] M. Li, S. Yu, Y. Zheng, K. Ren, and W. Lou, "Scalable and secure sharing of personal health records in cloud computing using attribute-based encryption," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 1, pp. 131–143, 2013.

[97] R. Lu, X. Lin, and X. Shen, "SPOC: A secure and privacy-preserving opportunistic computing framework for mobile-healthcare emergency," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 3, pp. 614–624, 2013.

[98] X. Liang, R. Lu, L. Chen, X. Lin, and X. Shen, "PEC: A privacy-preserving emergency call scheme for mobile healthcare social networks," *Journal of Communications and Networks*, vol. 13, no. 2, pp. 102–112, 2011.

[99] X. Liang, M. Barua, R. Lu, X. Lin, and X. S. Shen, "Healthshare: Achieving secure and privacy-preserving health information sharing through health social networks," *Computer Communications*, vol. 35, no. 15, pp. 1910–1920, 2012.

[100] K. Zhang, X. Liang, M. Baura, R. Lu, and X. S. Shen, "PHDA: A priority based health data aggregation with privacy preservation for cloud assisted wbans," *Information Sciences*, vol. 284, pp. 130–141, Elsevier, 2014.

[101] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Advances in Cryptology–EUROCRYPT 2005*. Springer, 2005, pp. 457–473.

[102] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *IEEE Symposium on Security and Privacy. SP'07.*, 2007, pp. 321–334.

[103] N. Attrapadung, B. Libert, and E. De Panafieu, "Expressive key-policy attribute-based encryption with constant-size ciphertexts," in *Public Key Cryptography–PKC 2011*. Springer, 2011, pp. 90–108.

[104] B. Waters, "Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization," in *Public Key Cryptography–PKC 2011*. Springer, 2011, pp. 53–70.

[105] V. Goyal, A. Jain, O. Pandey, and A. Sahai, "Bounded ciphertext policy attribute based encryption," in *Automata, Languages and Programming*. Springer, 2008, pp. 579–591.

[106] C. Chen, J. Chen, H. W. Lim, Z. Zhang, D. Feng, S. Ling, and H. Wang, "Fully secure attribute-based systems with short ciphertexts/signatures and threshold access structures," in *Topics in Cryptology–CT-RSA 2013*. Springer, 2013, pp. 50–67.

[107] A. Lewko, T. Okamoto, A. Sahai, K. Takashima, and B. Waters, "Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption," in *Advances in Cryptology–EUROCRYPT 2010*. Springer, 2010, pp. 62–91.

[108] A. Lewko and B. Waters, "Decentralizing attribute-based encryption," in *Advances in Cryptology–EUROCRYPT 2011*. Springer, 2011, pp. 568–588.

[109] N. Attrapadung and H. Imai, "Conjunctive broadcast and attribute-based encryption," in *Pairing-Based Cryptography–Pairing 2009*. Springer, 2009, pp. 248–265.

[110] K. Emura, A. Miyaji, A. Nomura, K. Omote, and M. Soshi, "A ciphertext-policy attribute-based encryption scheme with constant ciphertext length," in *Information Security Practice and Experience*. Springer, 2009, pp. 13–23.

[111] X. Li, D. Gu, Y. Ren, N. Ding, and K. Yuan, "Efficient ciphertext-policy attribute based encryption with hidden policy," in *Internet and Distributed Computing Systems*. Springer, 2012, pp. 146–159.

[112] J. Herranz, F. Laguillaumie, and C. Ràfols, "Constant size ciphertexts in threshold attribute-based encryption," in *Public Key Cryptography–PKC 2010*. Springer, 2010, pp. 19–34.

[113] N. Doshi and D. Jinwala, "Hidden access structure ciphertext policy attribute based encryption with constant length ciphertext," in *Advanced Computing, Networking and Security*. Springer, 2012, pp. 515–523.

[114] T. Nishide, K. Yoneyama, and K. Ohta, "Attribute-based encryption with partially hidden encryptor-specified access structures," in *Applied cryptography and network security*. Springer, 2008, pp. 111–129.

[115] J. Horwitz, "A survey of broadcast encryption," *Journal of ACM*, 2003.

[116] D. R. Stinson, "On some methods for unconditionally secure key distribution and broadcast encryption," in *Selected Areas in Cryptography*. Springer, 1997, pp. 3–31.

[117] D. R. Stinson and T. Van Trung, "Some new results on key distribution patterns and broadcast encryption," *Designs, Codes and Cryptography*, vol. 14, no. 3, pp. 261–279, 1998.

[118] M. Luby and J. Staddon, "Combinatorial bounds for broadcast encryption," in *Advances in Cryptology–EUROCRYPT'98*. Springer, 1998, pp. 512–526.

[119] C. Gentry and B. Waters, "Adaptive security in broadcast encryption systems (with short ciphertexts)," in *Advances in Cryptology-EUROCRYPT 2009*. Springer, 2009, pp. 171–188.

[120] A. Fiat and M. Naor, "Broadcast encryption," in *Advances in Cryptology-CRYPTO93*. Springer, 1994, pp. 480–491.

[121] D. Naor, M. Naor, and J. Lotspiech, "Revocation and tracing schemes for stateless receivers," in *Advances in Cryptology–CRYPTO 2001*. Springer, 2001, pp. 41–62.

[122] D. Halevy and A. Shamir, "The LSD broadcast encryption scheme," in *Advances in Cryptology–CRYPTO 2002*. Springer, 2002, pp. 47–60.

[123] M. T. Goodrich, J. Z. Sun, and R. Tamassia, "Efficient tree-based revocation in groups of low-state devices," in *Advances in Cryptology–CRYPTO 2004*. Springer, 2004, pp. 511–527.

[124] S. S. D. Selvi, S. S. Vivek, R. Gopalakrishnan, N. N. Karuturi, and C. P. Rangan, "Provably secure ID-Based Broadcast Signcryption (IBBSC) scheme," *IACR Cryptology ePrint Archive*, vol. 2008, p. 225, 2008.

[125] D. Boneh, C. Gentry, and B. Waters, "Collusion resistant broadcast encryption with short ciphertexts and private keys," in *Advances in Cryptology–CRYPTO 2005*. Springer, 2005, pp. 258–275.

[126] C. Delerablée, P. Paillier, and D. Pointcheval, "Fully collusion secure dynamic broadcast encryption with constant-size ciphertexts or decryption keys," in *Pairing-Based Cryptography–Pairing 2007*. Springer, 2007, pp. 39–59.

[127] D. Boneh and B. Waters, "A fully collusion resistant broadcast, trace, and revoke system," in *Proceedings of the 13th ACM Conference on Computer and Communications Security*, ser. CCS '06. New York, NY, USA: ACM, 2006, pp. 211–220.

[128] D. Boyd, "Public safety statement of requirement for communication and interoperability," Homeland Security (US), The SAFECOMM program, October 2006.

[129] "First responder mobile application development best practices guide," Department of Homeland Security, USA.

[130] N. Koblitz and A. J. Menezes, "The random oracle model: a twenty-year retrospective," *Designs, Codes and Cryptography*, vol. 77, no. 2-3, pp. 587–610, 2015.

[131] D. Bernhard, M. Fischlin, and B. Warinschi, "Adaptive proofs of knowledge in the random oracle model," *IET Information Security*, vol. 10, no. 6, pp. 319–331, 2016.

[132] J. R. Douceur, "The sybil attack," in *Peer-to-peer Systems*. Springer, 2002, pp. 251–260.

[133] Certicom, "Benefits of ECC on server performance", "https://www.certicom.com/images/pdfs/sun-ecc-server-performance.pdf," March 2013.

[134] M. Lesk, "Electronic medical records: Confidentiality, care, and epidemiology," *IEEE Security Privacy*, vol. 11, no. 6, pp. 19–24, Nov 2013.

[135] A. Broder and M. Mitzenmacher, "Network applications of bloom filters: A survey," *Internet mathematics*, vol. 1, no. 4, pp. 485–509, 2004.

[136] B. H. Bloom, "Space/time trade-offs in hash coding with allowable errors," *Communications of the ACM*, vol. 13, no. 7, pp. 422–426, 1970.

[137] K. Suzuki, D. Tonien, K. Kurosawa, and K. Toyota, "Birthday paradox for multi-collisions," in *Information Security and Cryptology–ICISC 2006*. Springer, 2006, pp. 29–40.

[138] M. Bellare, R. Canetti, and H. Krawczyk, *Keying Hash Functions for Message Authentication*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1996, pp. 1–15. [Online]. Available: http://dx.doi.org/10.1007/3-540-68697-5_1

[139] M. Bellare, *New Proofs for NMAC and HMAC: Security Without Collision-Resistance*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 602–619. [Online]. Available: http://dx.doi.org/10.1007/11818175_36

[140] M. Fischlin, *Security of NMAC and HMAC Based on Non-malleability*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 138–154. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-79263-5_9

[141] H. Perl, Y. Mohammed, M. Brenner, and M. Smith, "Privacy/performance trade-off in private search on bio-medical data," *Future Generation Computer Systems*, vol. 36, pp. 441–452, Elsevier, 2014.

[142] A. D. Brucker, H. Petritsch, and S. G. Weber, "Attribute-based encryption with break-glass," in *Information Security Theory and Practices. Security and Privacy of Pervasive Systems and Smart Devices*. Springer, 2010, pp. 237–244.

[143] A. Kate, G. Zaverucha, and I. Goldberg, "Pairing-based onion routing," in *Privacy enhancing technologies*. Springer, 2007, pp. 95–112.

[144] C. Adams, "A classification for privacy techniques," *University of Ottawa Law & Technology Journal*, vol. 3, no. 1, pp. 35–52, 2006.

[145] S. Capkun and J.-P. Hubaux, "Secure positioning of wireless devices with application to sensor networks," in *Proceedings IEEE INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies.*, vol. 3, March 2005, pp. 1917–1928 vol. 3.

[146] S. Capkun, K. Bonne Rasmussen, M. Cagalj, and M. Srivastava, "Secure location verification with hidden and mobile base stations," *IEEE Transactions on Mobile Computing*, vol. 7, no. 4, pp. 470–483, April 2008.

[147] R. M. Needham and M. D. Schroeder, "Using encryption for authentication in large networks of computers," *Communications of the ACM*, vol. 21, no. 12, pp. 993–999, 1978.

[148] M. Bellare, R. Canetti, and H. Krawczyk, "Keying hash functions for message authentication," in *Advances in Cryptology–CRYPTO96*. Springer, 1996, pp. 1–15.

[149] I. Butun, M. Erol-Kantarci, B. Kantarci, and H. Song, "Cloud-centric multi-level authentication as a service for secure public safety device networks," *IEEE Communications Magazine*, vol. 54, no. 4, pp. 47–53, April 2016.

[150] X. Liang, Z. Cao, H. Lin, and J. Shao, "Attribute based proxy re-encryption with delegating capabilities," in *Proceedings of the 4th International Symposium on Information, Computer, and Communications Security*, ser. ASIACCS '09. New York, NY, USA: ACM, 2009, pp. 276–286.

[151] S. Yu, C. Wang, K. Ren, and W. Lou, "Attribute based data sharing with attribute revocation," in *Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security*, ser. ASIACCS '10. New York, NY, USA: ACM, 2010, pp. 261–270.

[152] Y. Shucheng, W. Cong, R. Kui, and L. Wenjing, "Achieving secure, scalable, and fine-grained data access control in cloud computing," in *Proceedings IEEE INFOCOM*, March 2010, pp. 1–9.

[153] Space and Naval Warfare Systems Center Atlantic, "GPS blue force tracking systems application note," U.S. Department of Homeland Security (DHS), System Assessment and Validation for Emergency Responders (SAVER), 2014.

[154] Space and Naval Warfare Systems Center Atlantic, "Automatic vehicle locating systems-summary," U.S. Department of Homeland Security (DHS), System Assessment and Validation for Emergency Responders (SAVER), 2010.

[155] S. Kumar and E. H. Spafford, "A pattern matching model for misuse intrusion detection," The COAST Project, Dept. of Computer Sciences, Purdue University, West Lafayette, IN, USA, Tech. Rep. CSD-TR-94-013, June 1994.

[156] R. Canetti, S. Halevi, and J. Katz, "Chosen-ciphertext security from identity-based encryption," in *Advances in Cryptology-Eurocrypt 2004*. Springer, 2004, pp. 207–222.

[157] C. Chen, Z. Zhang, and D. Feng, "Efficient ciphertext policy attribute-based encryption with constant-size ciphertext and constant computation-cost," in *Provable Security*. Springer, 2011, pp. 84–101.

[158] Z. Zhou, D. Huang, and Z. Wang, "Efficient privacy-preserving ciphertext-policy attribute based-encryption and broadcast encryption," *IEEE Transactions on Computers*, vol. 64, no. 1, pp. 126–138, Jan 2015.

[159] Z. Wang, D. Huang, Y. Zhu, B. Li, and C.-J. Chung, "Efficient attribute-based comparable data access control," *IEEE Transactions on Computers*, vol. 64, no. 12, pp. 3430–3443, Dec 2015.

[160] Y. Zhang, D. Zheng, X. Chen, J. Li, and H. Li, "Computationally efficient ciphertext-policy attribute-based encryption with constant-size ciphertexts," in *Provable Security*. Springer, 2014, pp. 259–273.

[161] A. Ge, R. Zhang, C. Chen, C. Ma, and Z. Zhang, "Threshold ciphertext policy attribute-based encryption with constant size ciphertexts," in *Information Security and Privacy*. Springer, 2012, pp. 336–349.

[162] Y. S. Rao and R. Dutta, "Recipient anonymous ciphertext-policy attribute based encryption," in *Information Systems Security*. Springer, 2013, pp. 329–344.

[163] W. Teng, G. Zhang, Y. Xiang, and D. Wang, "Attribute-based access control with constant-size ciphertext in cloud computing," *IEEE Transactions on Cloud Computing*, vol. PP, no. 99, pp. 1–1, 2016.

[164] R. Roman, C. Alcaraz, and J. Lopez, "A survey of cryptographic primitives and implementations for hardware-constrained sensor network nodes," *Mobile Networks and Applications*, vol. 12, no. 4, pp. 231–244, 2007.

[165] L. B. Oliveira, D. F. Aranha, C. P. Gouvêa, M. Scott, D. F. Câmara, J. López, and R. Dahab, "Tinypbc: Pairings for authenticated identity-based non-interactive key distribution in sensor networks," *Computer Communications*, vol. 34, no. 3, pp. 485–493, 2011.

[166] Crypto++.(2009), "Crypto++5.6.0benchmark," Tech. Rep. [Online]. Available: http://www.cryptopp.com/benchmarks-amd64.html

[167] S. Hohenberger and B. Waters, "Attribute-based encryption with fast decryption," in *Public-Key Cryptography–PKC 2013*. Springer, 2013, pp. 162–179.

[168] D. M. Freeman, "Converting pairing-based cryptosystems from composite-order groups to prime-order groups," in *Advances in Cryptology–EUROCRYPT 2010*. Springer, 2010, pp. 44–61.

[169] A. Lewko, "Tools for simulating features of composite order bilinear groups in the prime order setting," in *Advances in Cryptology–EUROCRYPT 2012*. Springer, 2012, pp. 318–335.

[170] A. Guillevic, "Comparing the pairing efficiency over composite-order and prime-order elliptic curves," in *Applied Cryptography and Network Security*. Springer, 2013, pp. 357–372.

[171] "National Emergency Number Association," Last Visited, November 2016. [Online]. Available: http://www.nena.org/

[172] C. Grant, National Institute of Standards and Technology (NIST), Special publication 1174, Arlington, Virginia, US, March 2014.

[173] K. Govindan and P. Mohapatra, "Trust computations and trust dynamics in mobile adhoc networks: A survey," *IEEE Communications Surveys Tutorials*, vol. 14, no. 2, pp. 279–298, Second 2012.

[174] R. R. Hoffman, M. Johnson, J. M. Bradshaw, and A. Underbrink, "Trust in automation," *IEEE Intelligent Systems*, vol. 28, no. 1, pp. 84–88, Jan 2013.

[175] W. Sherchan, S. Nepal, and C. Paris, "A survey of trust in social networks," *ACM Comput. Surv.*, vol. 45, no. 4, pp. 47:1–47:33, Aug. 2013. [Online]. Available: http://doi.acm.org/10.1145/2501654.2501661

[176] A. Jøsang, R. Ismail, and C. Boyd, "A survey of trust and reputation systems for online service provision," *Decision support systems*, vol. 43, no. 2, pp. 618–644, 2007.

[177] "An emergency management framework for canada," Public Safety Canada, Emergency Management Policy Directorate, 2011.

APPENDIX

Confidence Interval Computation

A Confidence Interval (CI) is used to quantify the uncertainty in any collected sample of data. It is defined as the estimated range of values within which a generated data lies with a specified probability. Simulated results such as Route availability, end-to-end delay, and reliability are measured by taking the mean of a succession of n runs, with different simulation seeds to ensure that there is no correlation in the presented results. All simulation runs have the same environment (identical) although they are independent from each other.

For example, the result for the Buffer Overflow Probability (BOP) is considered. From the n independent results (i.e., $\beta_1, \beta_2, ..., \beta_n$), β_i represents the BOP for the defined scenario from simulation run i. The mean of all the simulation measurements is computed as in (7.1).

$$\overline{\beta} = \frac{1}{n} \sum_{i=1}^{n} \beta_i. \tag{7.1}$$

Considering the mean value in (7.1), it gives an estimate of the expected value of β (i.e., $E[\beta] = \phi$). To ensure that this value represents a very close approximation of the true mean, we also need to compute the variance σ^2. The variance can be calculated using (7.2).

$$\sigma^2 = \frac{1}{n-1}\sum_{i=1}^{n}\left(\beta_i - \overline{\beta}\right)^2 \tag{7.2}$$

A small σ^2 indicates that the results are tightly clustered around the mean value and we can be confident that the $\overline{\beta}$ is in fact close to the $E[\beta]$. However, for a large σ^2, the results are widely dispersed around the mean and we are less confident that $\overline{\beta}$ is close to the $E[\beta]$. Here, we can specify an interval of values that with high probability contain the true value of the parameter. This interval can be defined such that with 0.95 probability the true values are obtained as shown in (7.3).

$$\Pr[L(\beta) \leq \phi \leq U(\beta)] = 0.95 \tag{7.3}$$

This interval a is 95% Confidence Interval (CI). Since the number of measurements applied is less than 100, using the Standard deviation and $t-$distribution table, the boundaries of the interval are calculated as in (7.4) and (7.5).

$$L(\beta) = \overline{\beta} - t_{2.5,d_f}\frac{\sigma}{\sqrt{n}}, \tag{7.4}$$

$$U(\beta) = \overline{\beta} + t_{2.5,d_f}\frac{\sigma}{\sqrt{n}}, \tag{7.5}$$

where n is the number of measurements, $d_f = n - 1$ is the degree of freedom, and σ is the standard deviation of the measurements.