

Security Enhancement over Transient Social Network Mobile Application for Disaster Management

An M. Tech. Thesis

*Submitted in partial fulfillment of the requirements
for the degree of*

Master of Technology

by

Prateeksha Keshari

153050047

under the guidance of

Prof. R.K. Shyamasundar



Department of Computer Science Engineering
Indian Institute of Technology Bombay
July 2017

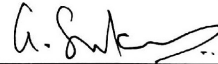
Dissertation Approval

This dissertation entitled **Security Enhancement Over Transient Social Network Mobile Application for Disaster Management** by Prateeksha Keshari (Roll Number: 153050047) is approved for the degree of Master of Technology in Computer Science and Engineering from IIT Bombay.

Examiners

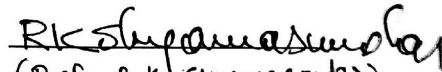


(Prof. Virendra Singh)



(Prof. G. Sivakumar)

Supervisor



(Prof. R.K. Shyamasundar)

Chairman



(Prof. Virendra Singh)

Date: 3 July 2017

Place: IIT BOMBAY

Declaration

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.



(Signature)

Prateeksha Keshari

153050047

Date: 3 July 2017

Acknowledgement

Foremost, I would like to express my gratitude to my advisor **Prof. R.K. Shyamasundar** for his continuous support in my M.Tech. project on *Security Enhancement over transient social network mobile application for disaster management*. He, along with Computer Science and Engineering Department IIT Bombay ensured that I have all the resources to carry out this work.

I would also like to thank **Dr. Vishwas Patil** for his continuous feedback in revising this thesis. Many thanks to CSIRO Data 61 team with whom I am collaborating under Australia India Strategic Research Fund (AISRF) program. Special thanks to **Raj Gaire**, **Hendra Wijaya** and **Surya Nepal**. I thank **Chigullapally Sriharsha**, IIT Kanpur for his help. Hearty thanks to **Prof. R. K. Ghosh**, IIT Kanpur for providing his guidance and valuable insights.

Abstract

Disaster events lead to loss of life as well as financial loss [11]. Hence, there is a need to leverage efficient and dynamically scalable ICT infrastructure to capture real-time data from multiple digital channels for an effective response. For serving this purpose **AlertApp**, an Android application is created by Data 61 team of CSIRO, IIT Bombay and IIT Kanpur under Emergency Service Awareness (ESA) program of Australia-India Strategic Research Fund (AISRF) [16]. It allows the registered users to receive real-time alerts and subscription to different categories of alerts. It also creates a Transient Social Network (TSN) and works as a peer-to-peer messaging application in the absence of Internet connection. AlertApp is capable of sending alert messages with or without GPS information. In disaster scenarios, there are a variety of stakeholders at various trust levels with whom a victim may or may not like to share private or specific location/other information. Our work caters these types of requirements.

In this dissertation, our contributions have been towards i) the setting of TSN for AlertApp using hotspots, ii) preserving the security and privacy of the underlying communications, iii) enhancing the user interface etc. For the security of communication, we have used the Readers Writers Flow Model (RWFM) setup. Further, we have enhanced the user interface using the "emoji" to enable the stakeholders to communicate easily. In the dissertation, we describe the design, implementation, and performance of the system.

Contents

List of Tables	vii
List of Figures	viii
1 Introduction	1
1.1 Problem Definition	4
1.2 Motivation	5
1.3 Our Contribution	8
2 Related Work	10
2.1 Architecture of TSN	11
2.1.1 Message Generator	12
2.1.2 Message Collector	12
2.1.3 Message Distributor	12
2.1.4 Local Mules	13
3 Security and Privacy of Data for Disaster Management	15
3.1 Information Flow Control	15
3.1.1 Bell-LaPadula Model	15
3.1.2 Biba Integrity Model	16
3.2 A Lattice Model of Secure Information flow	17
3.3 Readers-Writers Flow Model	18

4	Architecture of the System	20
5	Implementation	25
5.1	Example Scenario	26
5.2	Threat Scenario	28
5.3	Preventing Sybil Attack	28
5.4	Modifications done in AlertApp	28
5.5	Peer to Peer Messaging	32
5.6	Transient Social Network	35
5.7	Privacy and Security Using RWFM	37
5.7.1	Need for Downgrade	39
5.8	Test Results Under Different Environments	39
6	Conclusion and Future work	41
	References	44

List of Tables

3.1	Illustration why IFC is important	16
5.1	Test results	40

List of Figures

1.1	Tsunami [6]	1
1.2	Steps in disaster management [5]	3
1.3	Illustrative implementation of the disaster management application [16]	4
1.4	AlertApp: An Android application (2015) [16]	5
1.5	An illustration of change of route due to Sybil attack (Credit: Sinai et al)	6
1.6	Vulnerability during disaster	7
2.1	TSN architecture [18]	11
4.1	Islands of connected network	22
4.2	Maximum nodes are connected	23
4.3	Establishment of Delay Tolerant Network over Transient Social Network	24
5.1	The Framework of the AlertApp	25
5.2	Stampede scenario	27
5.3	Interface of modified AlertApp	30
5.4	Behaviour of modified AlertApp in a stampede alert	31
5.5	Peer to Peer messaging	33
5.6	GPS location received	34
5.7	Tab to create Transient Social Network	35

5.8	Distress node can send message now	37
-----	--	----

Chapter 1

Introduction

Our world faces many disasters, namely earthquakes, hurricanes, floods, and the threat of epidemics often, making us vulnerable to natural disaster [11]. This problem is further increased as we rely on computers, Internet communication, and these technologies stop working in the event of a disaster.



Figure 1.1: Tsunami [6]

The Tsunami of late 2004, Hurricanes Rita and Katrina, and the earth-

quake in Pakistan in 2005 has emphasized the importance of disaster management. They highlighted our ill-preparedness towards natural calamities. EM-DAT (2006) figures compiled by the Belgian Université Catholique de Louvain's Center for Research on the Epidemiology of Disasters (CRED) and the United Nations International Strategy for Disaster Reduction (UN/ISDR) indicate that:

- There were 360 natural disasters in 2005 compared to 305 natural disasters in 2004, and an increase of 18%. The number of droughts rose from 15 in the year 2004 to 22 in the year 2005, an increase of 47%. Floods too increased from 107 in the year 2004 to 168 in the year 2005, an increase of drastic 57%.
- About 157 million people were affected by disaster events, which was an increase of 7 million when compared to 2004. Though the death toll in 2004 was 244,500, which is very high as compared to 2005 in which the toll was 91,900. This is due to the tsunami which occurred in 2004.
- Total property worth \$159 billion was damaged in 2005 out of which hurricane Katrina caused a significant portion of \$125 billion. This is an increase of 71% as compared to the total loss in 2004, which was \$92.9 billion.

Though the death toll due to disasters has decreased over the last few decades, yet there is heavy loss of property and effect on development due to them. The risks of disasters need to be understood and proper management and mitigation strategies need to be planned. Investment should be made in resources so as to reduce the severity of disaster in terms of loss of life and loss of property. Technology has a major role to play in reducing the loss, especially the loss of life. This is where our contribution comes into the picture.

Given these changes and our increasing vulnerability, disaster management needs to consist of following steps:

1. Mitigation: In this phase, we aim to prevent the disaster and reduce the damage caused to property and loss of lives caused by them. Developing communication systems to work during a disaster, damage resistant data centers to prevent data loss, protection of hardware from damage are some of the examples of mitigation technique.
2. Preparedness: In the event of the disaster, measures need to be taken in order to reduce the loss of life and property. E.g. backing up data and securing servers from physical before the disaster strikes in and is an important part of disaster preparedness for many IT firms.

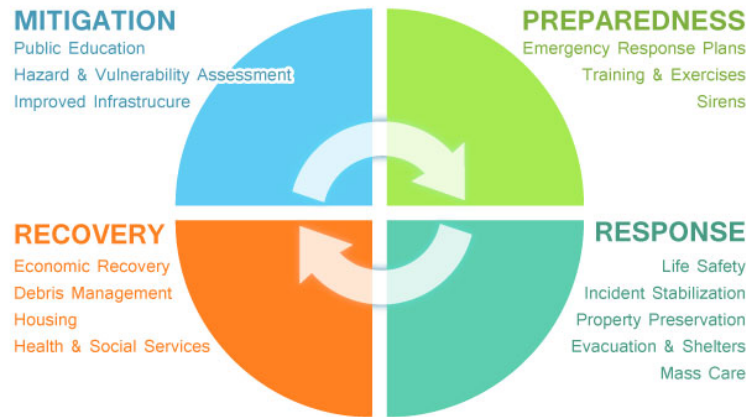


Figure 1.2: Steps in disaster management [5]

3. Emergency Response: During an event of the disaster, there is a need to rescue people who have been trapped or are in distress. This is the part where our AlertApp works. We provide a medium for communication through which people can communicate with each other, even if the Internet is not working through wifi links.
4. Recovery: The objective of the recovery phase is to eventually resume normal processing. It focuses on helping people to continue on with life, rebuilding and reestablishing infrastructures and businesses.

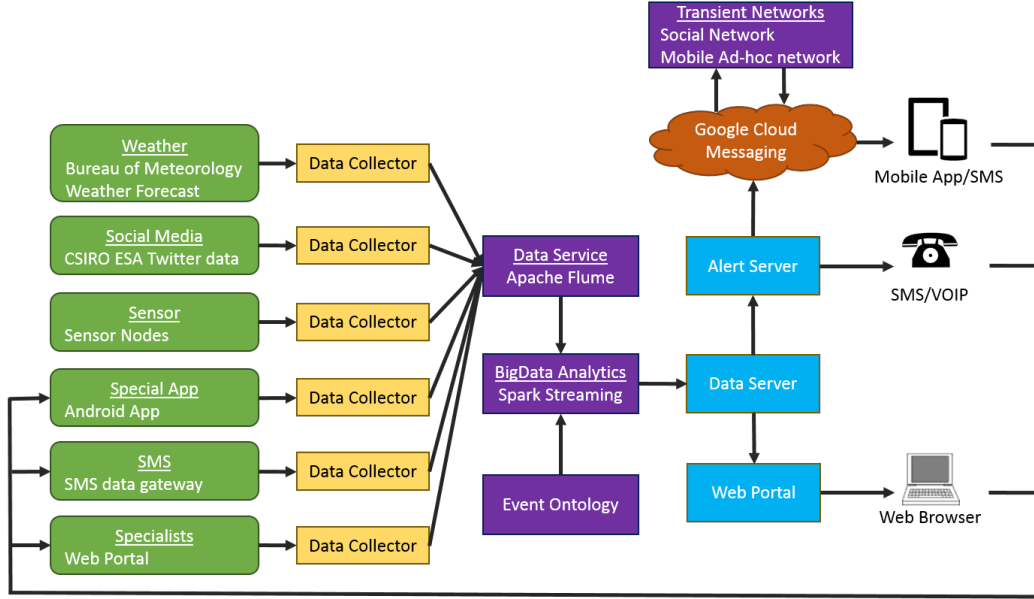


Figure 1.3: Illustrative implementation of the disaster management application [16]

1.1 Problem Definition

Disaster events all over the world leads to loss of life as well as financial loss [1] [2]. Effective response to crises and disaster events depends not only on the historical data, but also real-time data from multiple digital channels including social media feeds, text messages from mobile devices and sensor networks [16]. The architecture of disaster management application as shown in Figure 1.3 consists of many interfaces and many stakeholders, any user could peek into the contents of messages containing confidential information, or could hinder the functioning of the system in some way. In such system, a lattice model with labels of subject and objects is needed. The model should be able to implement proper information flow control, support multiple label changes at various stages. Readers-Writers Flow Model [14] satisfies these requirements and is hence applicable. Disasters lead to loss

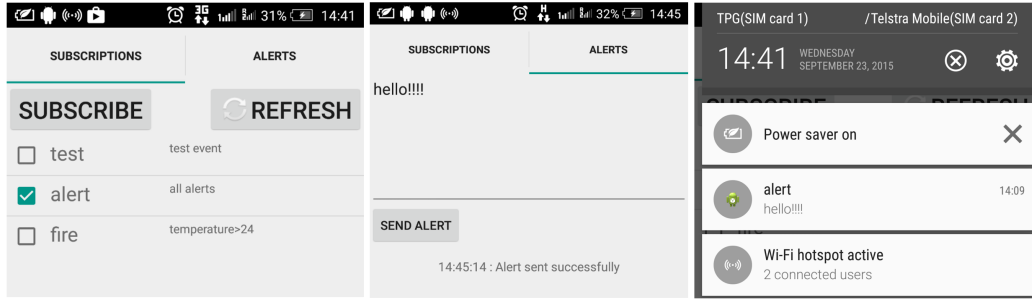


Figure 1.4: AlertApp: An Android application (2015) [16]

of infrastructure, more importantly, communication infrastructure. Communication with outside world is important for people in distress. We need a transient social network - it is an on-demand network in a local service zone. After building the TSN, a delay tolerant network (DTN) will be on top of it, so that as soon as connection reestablishes, the information can be sent to other service zones. The mobile application in Figure 2.1 is developed already, the server is set, the work on the sensors is also being done by the CSIRO. The work done is analysis where exactly we need to take care of message privacy, allowing only selected users to receive the message. Further Declassification of subjects(by the owner only) can be done. Implementing above mentioned point using Readers-Writers Flow Model. There is also an option to create Transient Social Network from within the AlertApp and a tab with peer to peer messaging service. Real-time location can too be sent by distress node to good Samaritan.

1.2 Motivation

The paper Exploiting Social Navigation [19] gives a detailed account of possible attacks on a social navigation Android application WAZE. The Sybil attack [19] is possible in a scenario of a disaster. The Sybil attack in computer

security is an attack wherein a reputation system is subverted by forging identities in peer-to-peer networks. This attack is due to lack of identity.

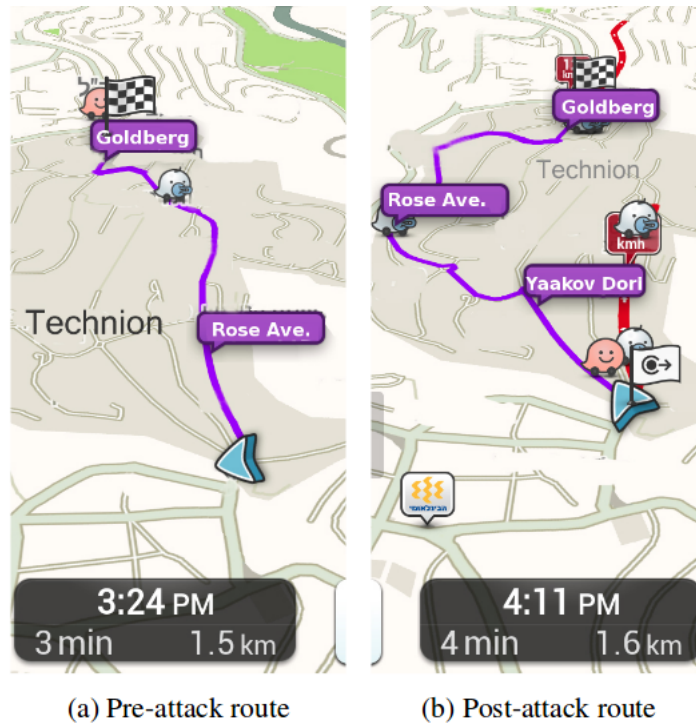


Figure 1.5: An illustration of change of route due to Sybil attack (Credit: Sinai et al)

The bots and malicious entities can simulate fake GPS report to influence social navigation systems, this is known as Sybil attack.

Figure 1.5 illustrates Sybil attack.

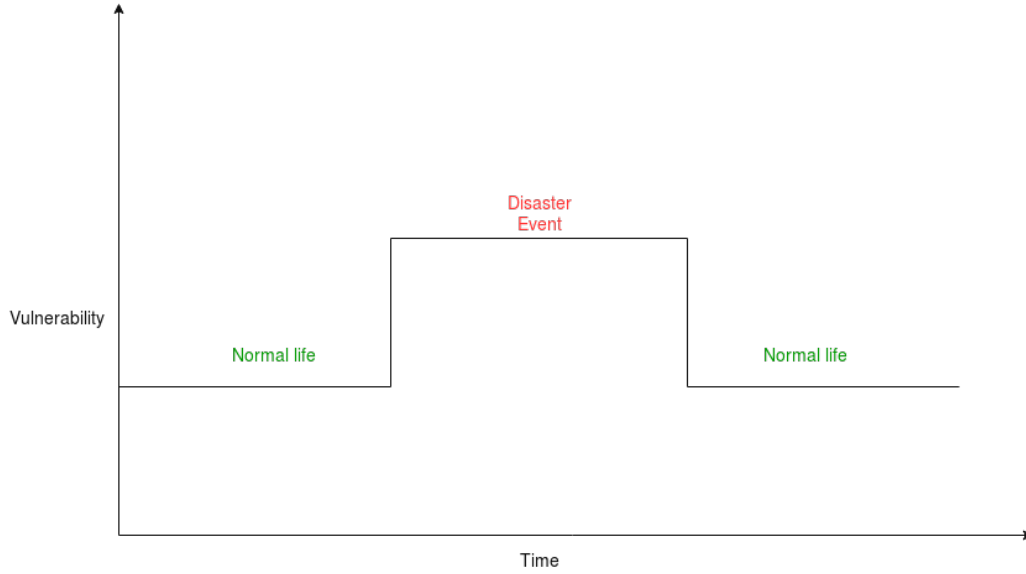


Figure 1.6: Vulnerability during disaster

Vulnerability increases in a disaster situation and people are willing to help the distressed persons. This vulnerability could be misused to perform a Sybil attack. The malicious user can simulate a fake disaster alarm(false alarm), the good Samaritan motivated to help can be caught in the attacker's web. There could be financial and security implications. For example, the Samaritan may be called to a lonely place for help and looted or even worse harmed. The similarity between WAZE and the AlertApp is in the fact that both works on crowd sourced data. Several attacks have huge security and financial implications. There is also a risk of the privacy breach in case of disclosure of user information. The attack demonstrated is Sybil attack, this attack is based on creating a large number of reputed "bot drivers", and controlling their reported locations using fake GPS reports. This attack can:

- Create false positive disaster conditions.
- Bombard the devices by false alerts.

- Can loot the Good Samaritan by calling him for help by showing mock location.
- Can divert attention by real disaster situation.

We need to prevent this attack. The earlier version of Alertapp was vulnerable.

1.3 Our Contribution

Our contributions are:

- Analyzing where exactly we need to take care about message privacy. There were lots of stakeholders at various trust levels with whom the victim may or may not like to share private or specific location/other information.
- Allowing only selected users to receive the message. In this work, we use Information Flow Control (IFC) to secure such privacy leaks.
- Declassification is also a feature that is provided.
- The information flow control model used to implement the above points is Readers-Writers Flow Model (RWFM). RWFM [14] [15] uses explicit readers and writers that provide a label model for capturing relationships and constraints of information flow among the stakeholders.
- A tab with peer to peer messaging service making it possible even if the Internet is not available.
- Option to create Transient Social Network from within the AlertApp. The device responsible for creating hotspot be data cart and other devices in an area can connect to it by joining the network. Regular communication can occur after the connection.

- Real-time location can too be sent by distress node to good Samaritan. It works even in absence of Internet via Global Positioning System (GPS), it also updates the location every 2 seconds.

Chapter 2

Related Work

The work by **Chigullapally Sriharsha** and **Prof R.K. Ghosh**, IIT Kanpur [18] was mainly focused in getting the message across during an event of the disaster without much focus on the privacy and integrity and authenticity of the message being communicated by the subjects. The work focused on creating a transient social network and then routing the messages so that they could reach others in case of major network failure due to disaster, namely floods and earthquakes. The routing relies on Delay Tolerant Network, or DTN. The basis of this network is the bundling of message and the technique of store and forward. The network works when the network is very erratic or maybe even unavailable. The nodes in the network stores the message in a queue (bundling) until the outgoing network is available. When the network is available, it sends all the messages in the queue. Hence, it works on the store and forward technique.

The TSN created is also based on android phones and uses wifi messages to communicate between the nodes since the network link most likely will not be available during the time of the disaster. The TSN is created temporarily with each node having a specific responsibility predefined at the time of registration. This is slightly different from our case where we have no predefined role for any user and a user may assume responsibility in case

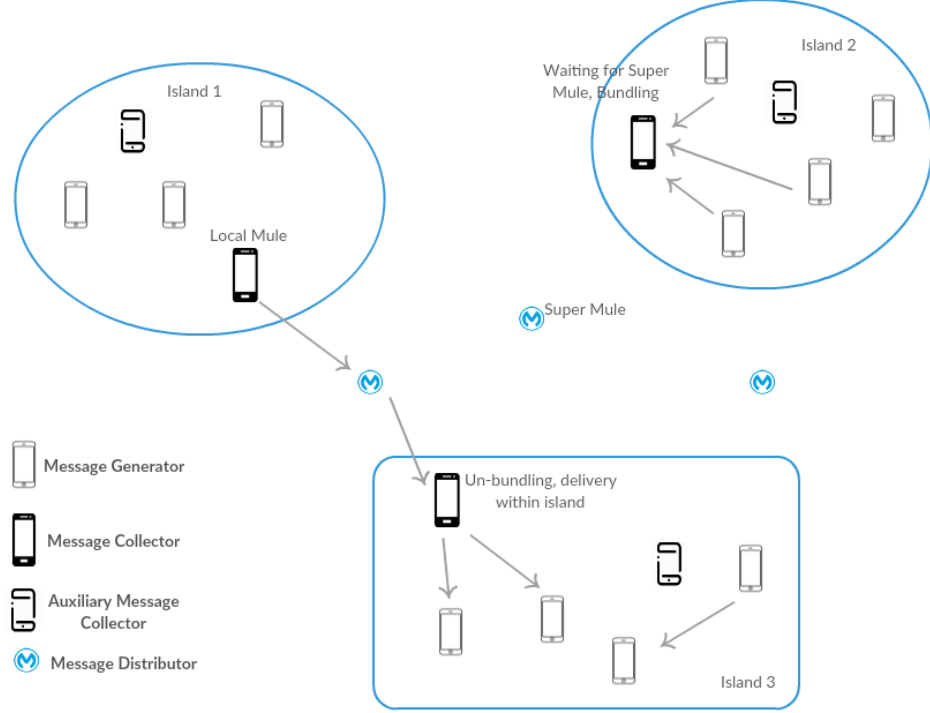


Figure 2.1: TSN architecture [18]

of disaster according to the condition.

2.1 Architecture of TSN

A TSN network contains message generators, message collector and message distributor. During an event of disaster, the backbone network such as Internet or cellular network gets disrupted. In such a scenario, the nearby nodes form a TSN based on wifi hotspot in order to exchange messages. Many such small TSN will emerge, forming various disconnected islands. Intra-island communication is made possible, but inter island communication is disrupted. To facilitate inter-island communication, the roles of nodes are

defined.

2.1.1 Message Generator

These are nodes which want to send message across. The message destination could be inside the same island or in another disconnected island. A distress node can be termed as a message generator.

2.1.2 Message Collector

Since the Internet or cellular network is unavailable, we need a node that can collect the messages until the messages can get delivered to the other islands. This here is done by a local mule node which bundles the messages sent by the generators. The collector also receives messages from the distributor and then disseminates the message to their respective recipients. In case the cellular network or DTN is working, the collector will also send the message using that network. There can be an auxiliary collector which will be functional as message collector when an existing message collector goes offline.

2.1.3 Message Distributor

This role is assumed by the super mules which move between the disconnected islands. A collector will transfer the message bundle to a distributor whenever a distributor comes in range of communication (wifi range). The distributor then receives all the messages of an island from the collector and then moves into the range of another collector of another island. This second collector will receive the messages from the distributor, unbundle the messages since it has received messages as a bundle. After unbundling, the collector disseminates the message to the respective target.

2.1.4 Local Mules

They collect messages from across the island and dump them to the message collector.

A TSN is set-up by a local mule by creating a wifi hotspot. All the distress nodes can connect to the mule and transfer the messages to it. The local mule will bundle up the messages. It is to be noted that this mule may be different from the node creating the TSN, but that case has not been handled in this work.

Whenever a super mule comes in range of the wifi hotspot created by the local mule, it will receive the messages bundled by the mule. The local mule also receives the message bundled up at the super mule, then spreads the message to the recipients after unbundling. A local mule also checks if DTN or cellular network is available. In case the network is available, it will send the network across using that network itself.

This architecture protocol is slightly different from our architecture protocol. The architecture has roles defined for various nodes, including mule and super mule. In our protocol, the TSN is not bound by the range of the wifi. Our TSN is defined by loosely connected nodes that can interact with each other using wifi. This architecture increases the range of TSN since by just using hotspot, the range is limited by the hotspot span of the central node (local mule in this case). But in our case, the nodes can move around and communicate via P2P message, hence there is no hard range defined for the TSN while also increasing the range to which a message can be communicated.

Moreover, we have no roles defined in case of our TSN. In our proposal, a node can receive a message, bundle it, and then transfer it to another node which is within range along with its own message (subject to the receiver list of both the nodes). Hence, there can be several local and super mules that can operate in a region, increasing the communication rate. Additionally,

we have focussed on authenticity of the messages by implementing RWFm privacy protocol. This is particularly necessary to avoid false alarms in case of disasters.

Chapter 3

Security and Privacy of Data for Disaster Management

3.1 Information Flow Control

Access control is the authentication of the subject which can access information, but not what action can subject perform with them. This can cause *leakage* of information. Many times such leaks happen, not because of defective *Access control* but when there is a lack of consideration of all possible leakage in Information Flow Control. Flow Controls decide the proper channel along which information can flow. In following sections we will describe the various information control models.

3.1.1 Bell-LaPadula Model

The table 3.1 shows Military based classification used in the famous Bell-LaPadula Model[7][8]. Even after being granted with access control, the top level information shouldn't flow to the level below or else it can cause security blunders.

It is also characterized by: "**no write down, no read up**" Let $SC(S) =$

Classification	Clearance	Objects
TOP SECRET (TS)	President	Nuclear Bomb Location File
SECRET (S)	User	Password File
CONFIDENTIAL (C)	Bidder	Bid Price File
UNCLASSIFIED (UC)	Citizens	Govt. Schemes File

Table 3.1: Illustration why IFC is important

q_s be security clearance of Subject S and $SC(O) = q_o$ be security classification of Object O .

There are two basic properties[9] regarding the read and write -

1. ***Simple Security Condition, Preliminary Version:***

S can read O if and only if $q_o \leq q_s$ and S has discretionary read access to O.

2. ****-Property (Star Property), Preliminary Version:***

S can write O if and only if $q_s \leq q_o$ and S has discretionary write access to O.

If these two properties are satisfied, the system is said to be secure.

3.1.2 Biba Integrity Model

- In 1975, Biba studied integrity of system.[13]
- It deal with **integrity** rather than **confidentiality**
- *Trustworthiness* is used as a measure of **integrity**.
- Characterized by the phrase: "no read down, no write up".[3]
- In contrast to the Bell-LaPadula model which is characterized by the phrase "no write down, no read up".
- Biba Integrity Model is a **dual** of bell-LaPadula model.

3.2 A Lattice Model of Secure Information flow

This model [10] is based on a mathematical framework capable of devising secure information flows within different security classes. At the heart of this model is a lattice structure made up of different security classes and rationalized by the semantics of information flow - like what information flow is permissible in given direction between security classes.

An *information flow model* is defined by:

$$\mathbf{IFM} = \langle N, P, SC, \oplus, \longrightarrow \rangle$$

- $N = \{a, b, \dots\}$ is a set of logical storage Objects.
- $P = \{p, q, \dots\}$ is a set of processes, they are responsible for information flow.
- $SC = \{A, B, \dots\}$ is a set of *security classes*, corresponding to disjoint classes of information.
- \oplus is the *class-combining operator*, associative and commutative in nature. This binary operator specifies the class in which the result of binary function belongs.
- \longrightarrow is a *flow relation* defined on pairs of security classes.

An information flow policy is defined by a lattice $\langle SC, \longrightarrow \rangle$ and hence $\langle SC, \longrightarrow \rangle$ is a *partial ordered set* therefore, these three properties holds -

1. $A \longrightarrow A$ (reflexive).
2. $A \longrightarrow B$ and $B \longrightarrow C \Rightarrow A \longrightarrow C$ (transitive).
3. $A \longrightarrow B$ and $B \longrightarrow A \Rightarrow A = B$ (antisymmetric).

3.3 Readers-Writers Flow Model

Readers-Writers Flow Model (RWFM) [14] [15] uses explicit readers and writers that provides a label model for capturing relationships and constraints of information flow among the stakeholders.

RWFM is defined as a five-tuple $\langle \mathbf{S}, \mathbf{O}, \mathbf{S} \times 2^S \times 2^S, (*, \supseteq, \subseteq), (*, \cap, \cup) \rangle$

- \mathbf{S} and \mathbf{O} denote the set of subjects and objects in the information system respectively
- $\mathbf{S} \times 2^S \times 2^S$ is the set of security labels
- $(*, \supseteq, \subseteq)$ denotes the can-flow-to ordering amongst labels
- $(*, \cap, \cup)$ denotes the label combining operator.

The first component of a RWFM label denotes the ownership of information, the second component denotes the permissible readers of information, and the third component denotes the set of principals that influenced the information.

$(s, S, \{s_f\})$ and $(s, \{s_f\}, S)$ are the "default label" and "clearance" for a subject s respectively, where S is the set of all the subjects in the system. Clearance is the highest level beyond which subject should not be permitted.

RWFM describes the conditions under which an operation is considered safe:

1. **READ Rule**

Subject s with label (s_1, R_1, W_1) requests read access to an object o with label (s_2, R_2, W_2) .

If $(s \in R_2)$ then

change the label of s to $(s_1, R_1 \cap R_2, W_1 \cup W_2)$

ALLOW

Else

DENY

2. **WRITE Rule**

Subject s with label (s_1, R_1, W_1) requests write access to an object o with label (s_2, R_2, W_2) .

If $(s \in W_2 \wedge R_1 \supseteq R_2 \wedge W_1 \subseteq W_2)$ then

ALLOW

Else

DENY

3. **DOWNGRADE Rule**

Subject s with label (s_1, R_1, W_1) requests to downgrade an object o from its current label (s_2, R_2, W_2) to (s_3, R_3, W_3) .

If $(s \in R_2 \wedge s_1 = s_2 = s_3 \wedge R_1 = R_2 \wedge W_1 = W_2 = W_3 \wedge R_2 \subseteq R_3 \wedge (W_1 = \{s_1\} \vee (R_3 - R_2 \subseteq W_2)))$ then

ALLOW

Else

DENY

4. **CREATE Rule**

Subject s with label (s_1, R_1, W_1) requests to create an object o . Create a new object o , label it as (s_1, R_1, W_1) and add it to the set of objects O .

Chapter 4

Architecture of the System

Following are the system components as discussed in the paper A Framework of Community Inspired Distributed Message Dissemination and Emergency Alert Response System over Smart Phones[12]:

1. Data Cart Tracker (DCT) - A Data Cart Tracker tracks information about the Data Carts for those SZ whose centers lie within the Data Cart Tracker's coverage area.
2. Primary Data Cart (PDC) - A Data Cart stores contact information of good Samaritans local to its own service zone. There is only one PDC for an SZ. (Message collector)
3. Auxiliary Data Cart (ADC) - There should be at least two Auxiliary Data Carts per SZ. Depending on the amount of traffic, the Primary Data Cart can choose to create more ADCs. (Auxiliary message collector)
4. Service Zone - A Service Zone is basically a contiguous part of GSM coverage area defining a community for localizing coordination of emergency response Activities.

5. Mule to act as AP for the wireless network on an island.

Following terms stand for:

- GS - Good Samaritan

Hence, there is an immediate need to leverage efficient and dynamically scalable ICT infrastructure as shown in Figure 1.3. Android mobile application is shown in Figure 2.1 has been developed allowing users to register and subscribe to alerts. In an emergency situation, relevant alerts are then sent to the app.

- DN - Distress Node
- RN - Node offering Resources RN

The Figure 4.1 shows the TSNs - island of connected nodes and now the island should be connected with DTN, but the black nodes are still not connected to any islands. The solution of this problem - Mules move around so that maximum possible nodes can be connected in an acceptable delay. Shown in the Figure 4.2.

After building the TSN, a delay tolerant network (DTN) will be on top of it, so that as soon as connection reestablishes, the information can be sent to other service zones. Figure 4.3 shows a Delay Tolerant Network.

We require a Distributed System where nodes can drop in and out and there is a dynamic one-to-one connection among nodes, we will focus on No dedicated node as sender, collector, distributor; Anyone can become mule and all the messages are distributed to everyone.

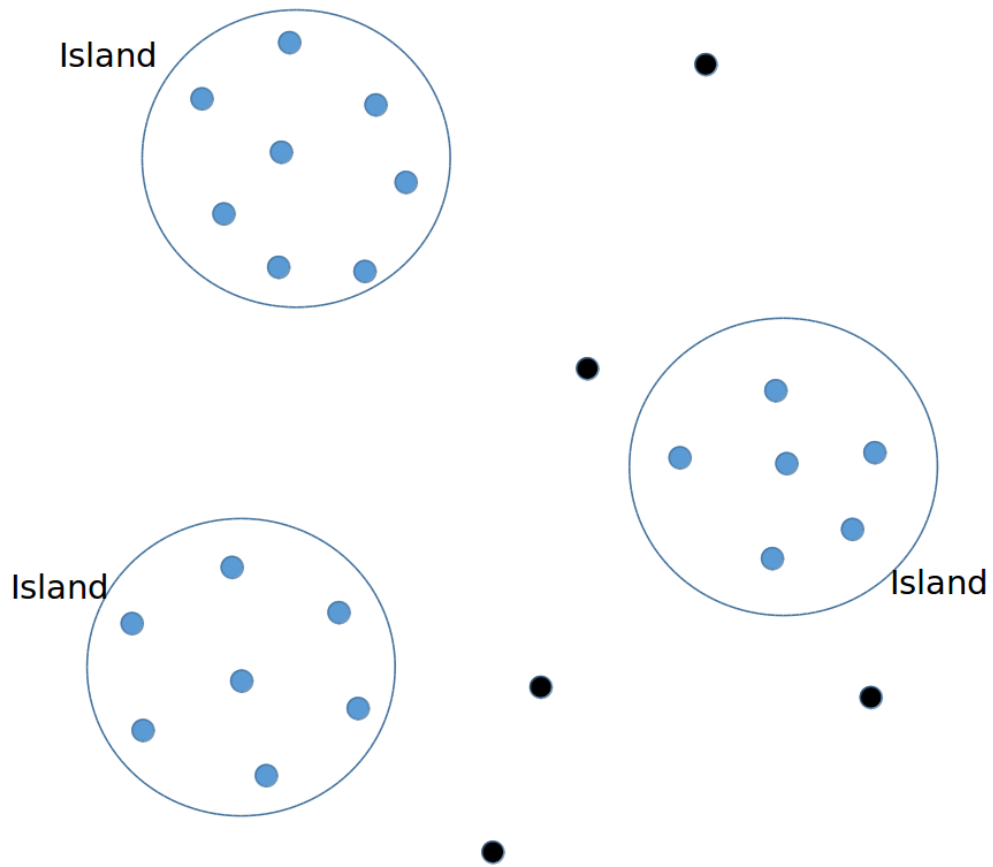


Figure 4.1: Islands of connected network

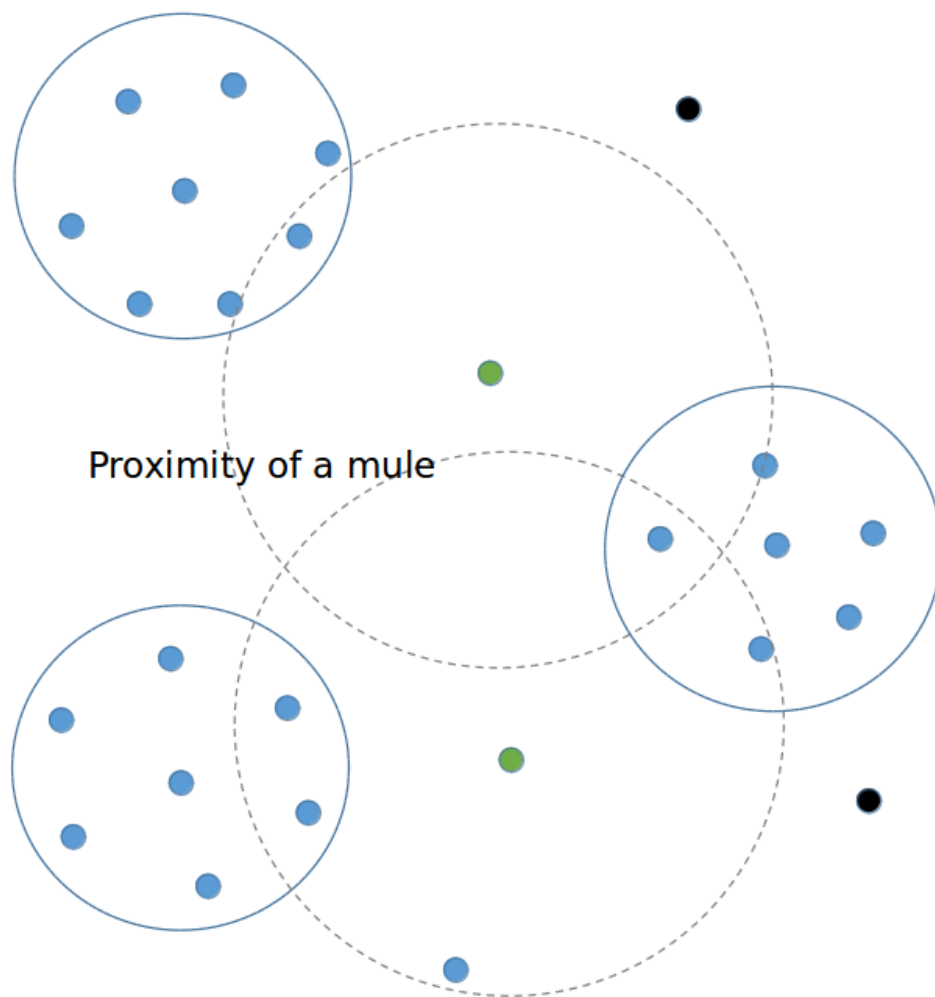


Figure 4.2: Maximum nodes are connected

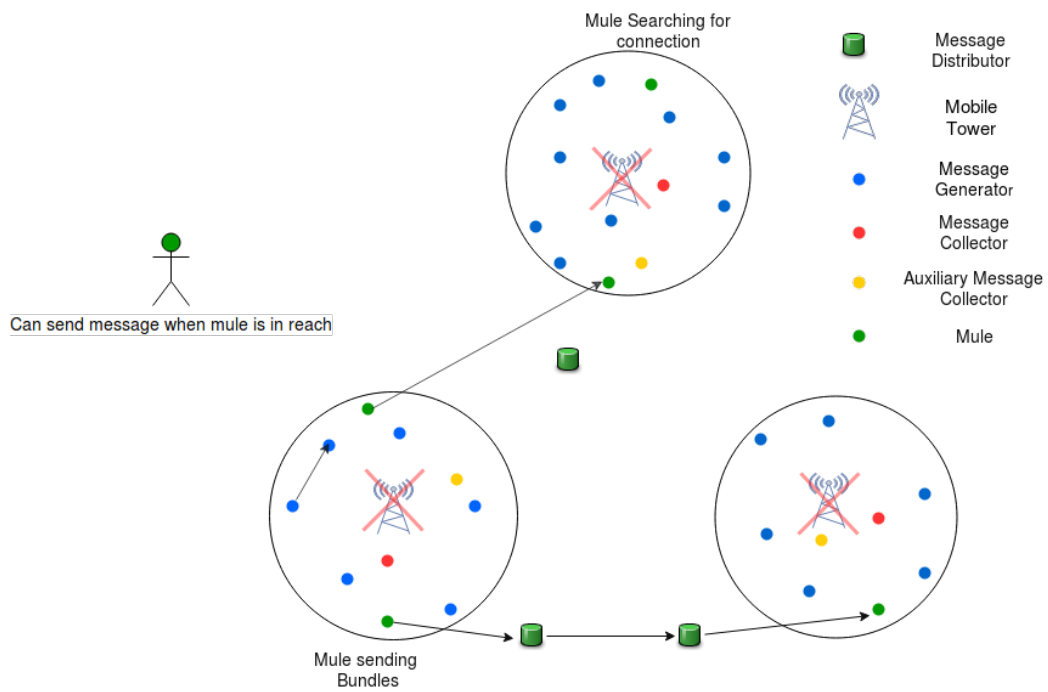


Figure 4.3: Establishment of Delay Tolerant Network over Transient Social Network

Chapter 5

Implementation

The work done is analysis where exactly we need to take care of message privacy, allowing only selected users to receive the message. Further declassification of subjects (by the owner only) can be done. Implementing above mentioned point using Readers-Writers Flow Model. The Figure 5.1 shows the features offered by the AlertApp.

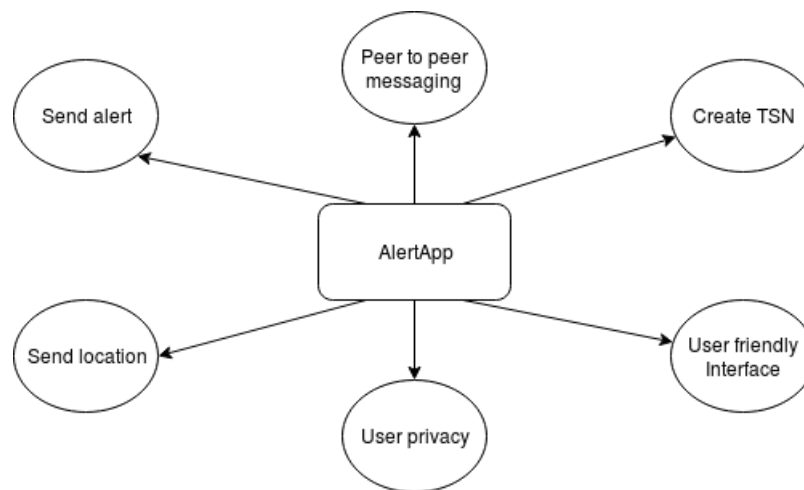


Figure 5.1: The Framework of the AlertApp

There is also an option to create Transient Social Network from within the

AlertApp and a tab with peer to peer messaging service. Real-time location can too be sent by distress node to good Samaritan.

In subsequent sections, the work will be well elaborated.

5.1 Example Scenario

Consider a scenario:

1. A good Samaritan wants to alert the authorities for possible stampede chances in a pilgrimage area, he may not want to broadcast the message, if he broadcasts the message there may be a chance that hearing about the chances, people hurry up and may lead to an actual stampede where there was just a chance.
2. Consider an example of a locality, there is an incidence of fire, the onlookers may send an alert message but as it is a broadcast, it may reach in wrong hands, like thieves, they can take advantage of the chaos and can steal the belongings.
3. Rescue operation takes place in steps. If there is a rescue operation, authorities may want to send the details about their operation only to people living in a locality and not actually broadcast it, because broadcasting the alert about it may cause chaos.

Figure 5.2 demonstrates the possible chaos if an alert about stampede possibility is broadcast: it may actually cause a stampede. In all the above scenarios we want that there must be **specific set of users** that can actually access that information. In this chapter, we will demonstrate that how we have achieved it.

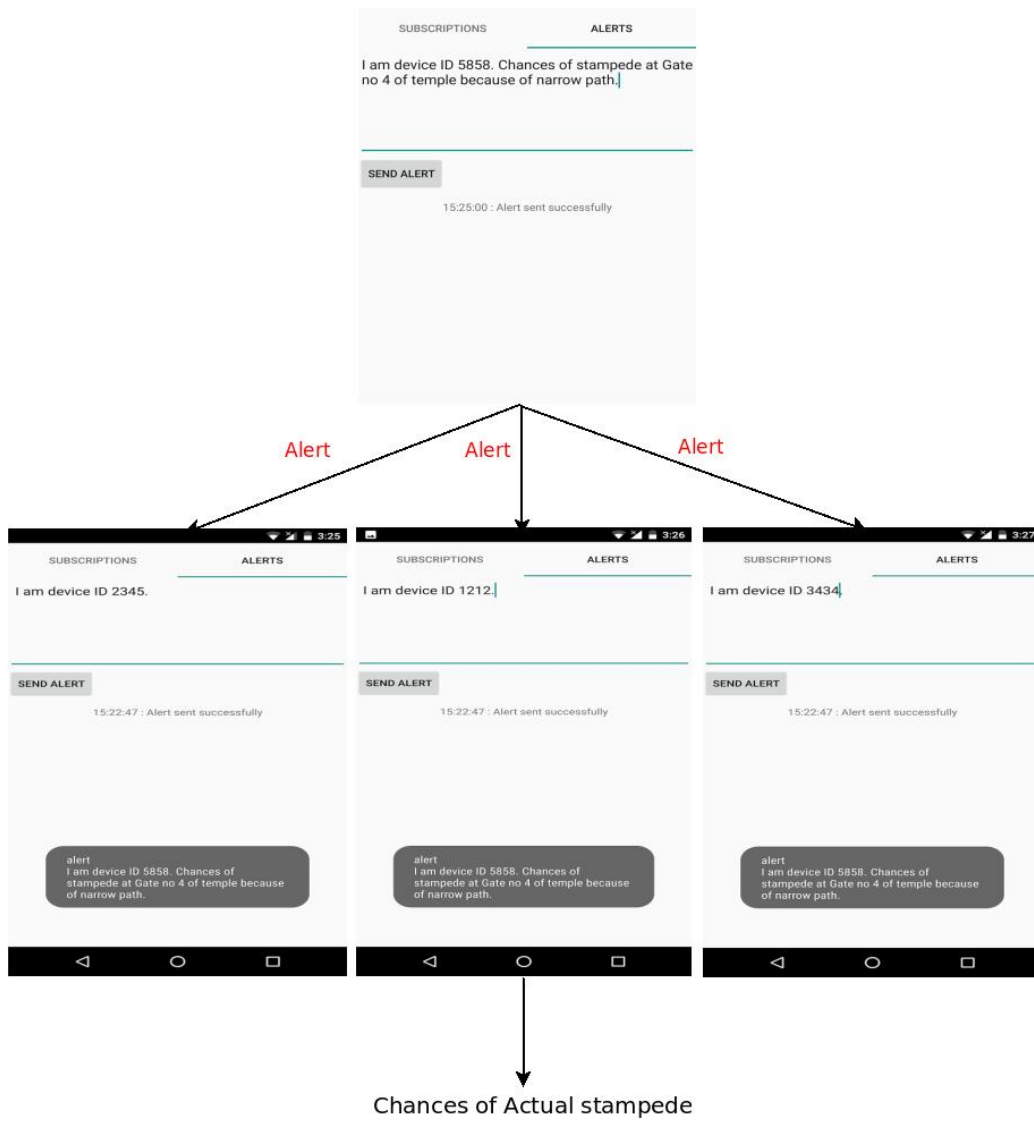


Figure 5.2: Stampede scenario

5.2 Threat Scenario

As discussed earlier, vulnerability increases in a disaster situation and people are willing to help the distressed persons. This vulnerability could be misused to perform a Sybil attack. The malicious user can simulate a fake disaster alarm (false alarm), the good Samaritan motivated to help can be caught in the attacker's web. There could be financial and security implications. For example, the Samaritan may be called to a lonely place for help and looted or even worse, harmed.

5.3 Preventing Sybil Attack

The main reason why Sybil attack is possible is because of lack of identity of the device. The Android platform manual[4] suggests application may need to identify a device rather than an instance of the application or an authenticated user on the device.

IMEI is an example of a unique identifier that could be used because in these scenarios, software IDs can be reset to avoid detection, so hardware identifiers may be used.

5.4 Modifications done in AlertApp

The work related to AlertApp is done in Android Studio.

A RWFM.java class is created which is in essence implementation of Readers-Writers Flow Model :

```
package au.csiro.ict.alertapp;
import static au.csiro.ict.alertapp.MainActivity.dev;

public class RWFM // Readers-Writers Flow model
{
```

```

private String reader="";
private String writer="";
private String subject=dev; //Initially subject is the owner
    itself, i.e device ID here

void add_reader(String s)
{
    reader=reader+"@"+s; //Readers will be appended, @ sign in
        between
}

String getReader()
{
    return reader;
}

void add_subject(String s)
{
    subject = subject + "@" +s;
    add_writer(s); // If new subjects/owners are added, they are
        automatically writer of/to that ID
}

String getSubject()
{
    return subject;
}

public void add_writer(String s)
{
    writer = writer + "@" + s;

```

```

    }

String getWriter()
{
    return writer;
}

}

```

Figure 5.3 displays how the modified app looks like. Note the additional functionality of **ADD SENDER** button. The device ID added here is inserted into reader's set of our Device ID.

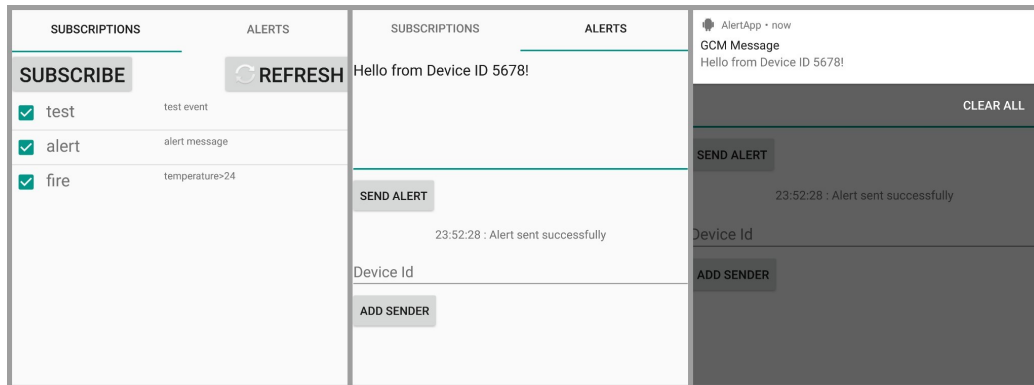


Figure 5.3: Interface of modified AlertApp

Figure 5.4 demonstrates what will happen when the stampede alert scenario is handled in modified AlertApp.

This service is used to push alerts over the Internet to other app users. It uses Google Cloud messaging service to send the messages and alerts. If a user has subscribed to the subscribe function (using Subscription button in the app), he will also get push notifications from other users. The function implements RWFm for privacy protection.

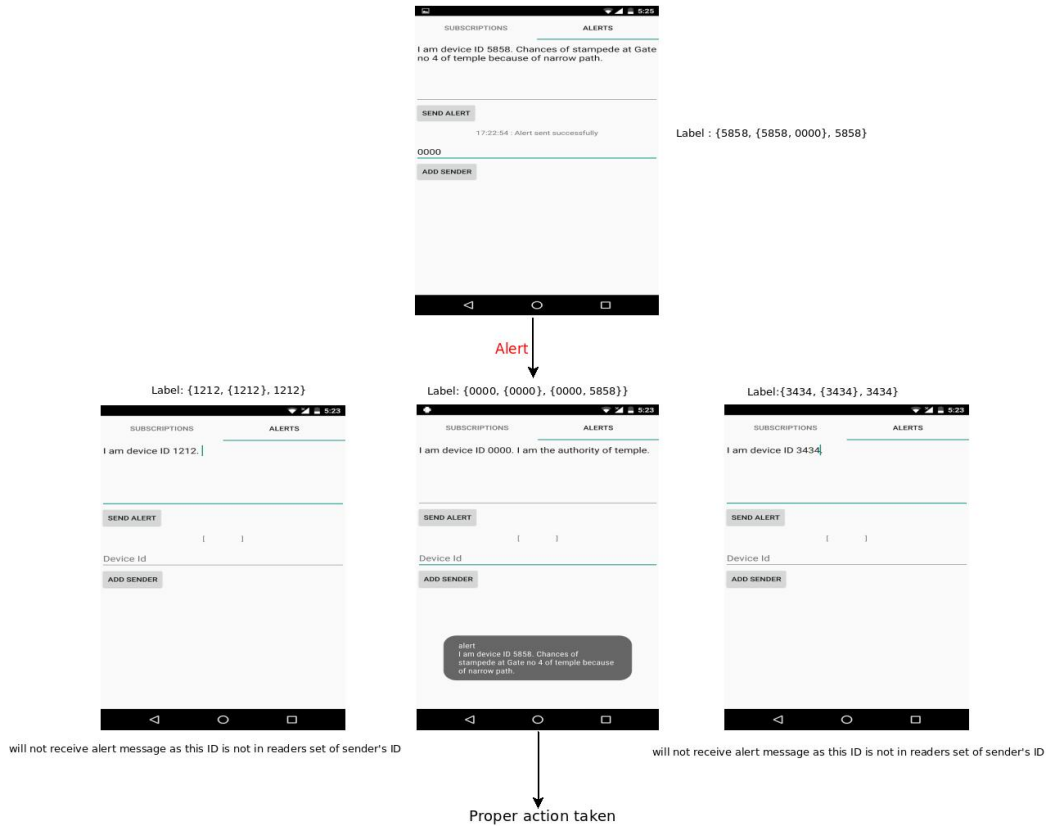


Figure 5.4: Behaviour of modified AlertApp in a stampede alert

The distress node can create his own message and send it to other users. The subscribed user will get a notification. If the sender specifies a receiver, that receiver will be added to the Reader's list of the sender and will receive the updates from the sender. Any other reader not in the Reader's list of the sender will not receive update even if they have subscribed to the service.

Another function provided is Declassify. As per RWFm, declassification refers to adding the subjects from the Writer list to the Reader list. This has the effect of reducing the privacy constraints. If a subject has ever written to the device (sent a notification to the device), that subject is entered into Writer's list of the receiving device. Hence, all the subjects who have to

send a notification to the device will be in its Writer's list. The Reader's list of the device contains all the subjects that can read the messages by the device, which is usually specified by the user. Declassify function updates the Reader's list of the device by adding all the subjects in the Writer's list to the Reader's list. Hence, all the subjects who have ever send a message to the device can now receive the updates from the device. This is equivalent to downgrade rule of RWFM: Subject s with label (s_1, R_1, W_1) requests to downgrade an object o from its current label (s_2, R_2, W_2) to (s_3, R_3, W_3) .

If $(s \in R_2 \wedge s_1 = s_2 = s_3 \wedge R_1 = R_2 \wedge W_1 = W_2 = W_3 \wedge R_2 \subseteq R_3 \wedge (W_1 = \{s_1\} \vee (R_3 - R_2 \subseteq W_2)))$ then

ALLOW

Else

DENY

5.5 Peer to Peer Messaging

In this mobile application, peer to peer messaging is also possible if the Internet is not available. It can also send the GPS location and alert message by just clicking a corresponding button. Once again privacy features can be added here.

P2P messaging is direct messaging between two entities without the involvement of any third entity server in between. This messaging uses wifi as a medium for communication. During a disaster, there are high chances that mobile Internet service may get disrupted. Hence, most of the current messaging apps are rendered useless. This is where our service of P2P messaging comes into play. Since all smartphones have wifi, this service will work regardless of mobile network condition.

This service has two functions, first, is "Start service" to start messaging service and second is "Discover service". The first function starts the service,

which indicates other that the mobile is ready to connect to a peer for message exchange. "Discover service" function searches for any service which has been activated. If a service is discovered, the mobile will be connected to the other mobile via wifi and transfer of message can take place. The connection by "Discover Service" function will connect to the first service detected unless specified otherwise by the user himself. This is particularly important in a disaster situation wherein a person in danger needs to get the message across as soon as possible so that he can be rescued. However, we have implemented the RWFM and the user (writer) can add readers to the reader list. This will ensure privacy as the "Discover Service" function will only connect to the specified readers.

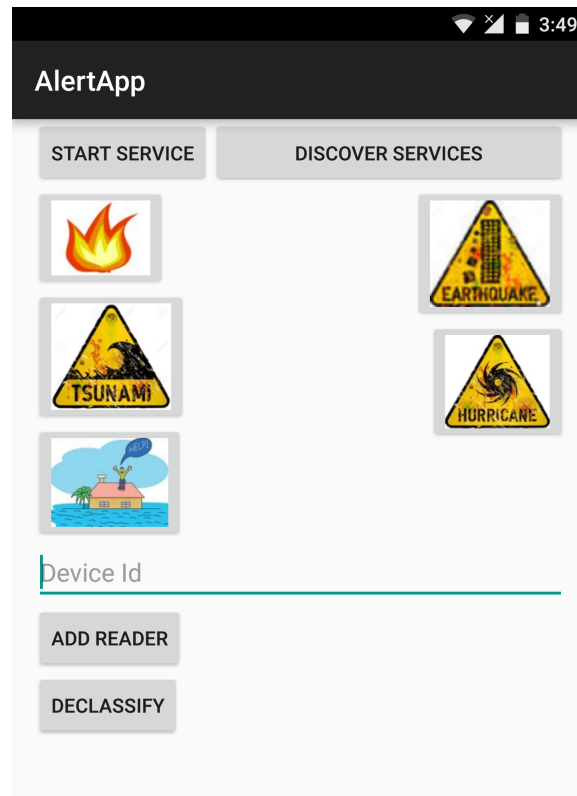


Figure 5.5: Peer to Peer messaging

The mobiles within the P2P range form an island in which entity can interact with each other as specified in Figure 4.1. Any rescuer within an island can contact the people who need help within the same island. But with the limited range of wifi, it is not possible to connect or send a message to entities of other islands directly.

We have defined message templates that could be sent directly with just a press of a button. Along with the message, the location of the person is also conveyed so that rescuer is able to get the location of the person. The message also triggers a map activity which shows the location of both rescuer and the person to be rescued, making the rescue operation easier and faster.



Figure 5.6: GPS location received

The GPS location is also sent when any of the button indicating disaster is clicked.

5.6 Transient Social Network

The AlertApp is capable of creating a Transient Social Network as shown in Figure 5.7 The device responsible for creating hotspot be data cart and other devices in the area can connect to it by Joining the network. The device sharing its Internet can be a Good Samaritan (GS). TSNApp is the name of the hotspot network.

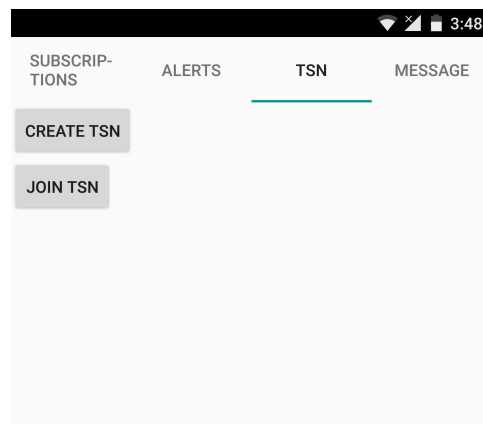


Figure 5.7: Tab to create Transient Social Network

Transient Social Network refers to a connection between multiple entities that are created by the requirement. Since we cannot rely on the Internet during the disaster, the connection between multiple entities must use either Bluetooth or wifi. Wifi is a better option since it has a larger range, though the power consumption is also higher.

A feature that TSN must support is communication with the outside world if and whenever possible. If there is an availability of network and Internet on any one device, it can be shared with other devices in the network

so that every device in the network is able to communicate using the Internet. This is done in our app through the use of hotspot and Internet sharing.

This feature consists of two functions. The first function is used to create a network. The device creating the network can be a Good Samaritan (GS). This function detects if there is any active wifi network that the device is connected to. If there is a wifi connection, the function disconnects from it and creates a new transient network.

The second feature is Join Network. This feature can be used by Distress node to connect to a transient network, if available in the vicinity, or else this feature will continue to search the vicinity until a TSN network is discovered. There is no role of the user of a device in establishing the connection. The only role that user has is to initiate the function to join a TSN. The search and network establishment are automatically done by the app. After establishing the TSN connection, the distress node can use the Internet if available at the node which has initiated the network. The alert using the Internet can be sent and received by all the nodes in the TSN. The designated nodes can also declassify/ downgrade the information, as shown in Figure 5.8.

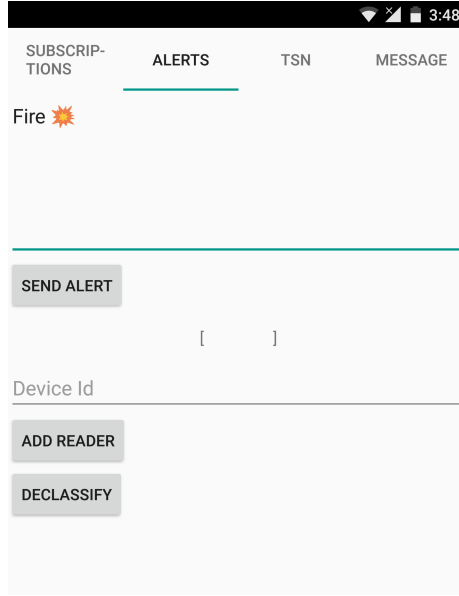


Figure 5.8: Distress node can send message now

5.7 Privacy and Security Using RWFM

A subject s_1 may be assigned a label (s_1, R, W) . Here, R is the set of the readers of s_1 who can read the messages sent by s_1 . W is the set of writers who have sent a message to s_1 . If a subject s_i is in writers list of s_1 , then s_1 will also be in the readers list of s_i .

The AlertApp initially broadcast the alert messages to all the users. This poses serious concerns as pointed out in the previous section. We implemented RWFM to address the privacy concerns.

Each AlertApp user s_i is defined by label (s_i, R_i, W_i) . Here, s_i can be the phone number of the user. Initially, both reader list R and writer list W are empty.

Now, the user can control the reader of a certain message by specifying the user in his reader's list. Let s_j be a user to whom s_i wants to send a message. Therefore, s_i will update his readers list by adding s_j to it.

$$R_i = R_i \cup s_j$$

Hence, by specifying the readers, we can control who receives the message. This part has been applied on alert messages of the AlertApp, so that only users specified in the Reader's list of the subject get the message.

We now explain the role of RWFM in peer to peer message. In P2P message, a distress node may need to send message to any node that is nearby, and there are chances that the distress node may not know the id of any nearby node. In such case, the user may use the broadcast message option to ask for help. However, RWFM comes into play when message needs to be sent beyond immediate neighbour node. In case of a disaster, the mobile facilities may be completely destroyed, therefore using P2P messages, we may still send messages across. By linking the chain of P2P messages which we aim to implement, we can increase the range of these messages. A subject s_i will send a message to s_j . Subject s_j acts as the mule node and later may pass on this message to s_k and so on. Here, if we do not restrict the readers, then s_i has no control over who could receive its message. Here, RWFM makes sure that only trusted readers can receive the messages. The subject s_k will receive message from s_j only if following conditions are met:

$$s_k \in R_i \cap R_j$$

This means s_k has to be in readers list of both s_i and s_j . Since all subjects are able to move, all subjects can act as distress node and mule at the same time.

5.7.1 Need for Downgrade

A subject s_i can downgraded itself to send message to more people. In this scenario, all the subjects in writer's list are added to its readers list. Hence, all the subjects that have send message to s_i in the past can receive the message from s_i .

The subject with label (s_i, R_i, W_i) will be downgraded to (s_j, R_j, W_j) such that:

$$(s_i \in R_j \wedge s_i = s_j \wedge W_i = W_j \wedge R_i \subseteq R_j \wedge (R_j - R_i \subseteq W_j))$$

Since the new subjects added to the reader's list were already present in the writer's list, it is reasonable to assume that sending message to them is safe. This is particularly necessary of exchange of messages from both side. A subject s_i may receive a message from another subject s_j . This implies that s_i is on the reader's list of s_j and the s_j will be added to the writer's list of s_i . In order to communicate back with s_j , s_i must add it to its reader's list. This can either be done by explicitly adding s_j to the reader's list of s_i , or by downgrading (downgrading adds all subjects from writer's list to reader's list).

5.8 Test Results Under Different Environments

We tested the AlertApp in different environment, namely indoors (inside the rooms) where the two devices were not in line of sight, in the corridors where devices were in line of sight and also the outdoor environment, where the devices were in line of sight but there were obstacles like trees, cars and lamp posts in between. The testing was done for TSN, P2P connection and message exchange. Since alert messages use the Internet, they are independent of the distance between the devices.

Indoors:

The devices were able to connect to TSN within the range of 15-20 meters.

Location	Range of connection establishment (in m)	Range of message transfer (in m)
Indoors	15-20	20-25
Corridors	35-40	70-80
Outdoors	35-40	80-90

Table 5.1: Test results

The P2P connection was also established within the same range. However, once the devices were connected in the P2P connection, the message could be exchanged within the range of 20-25 meters.

Corridors:

When the two devices are in a line of sight, their range of connection and message exchange increased considerably. The TSN was discovered and a connection was established within the range of 35-40 meters. The P2P connection was also established within the range of 40 meters. The message could be exchanged withing good 70 to 80 meters once the connection was established.

Outdoors:

The result of the app in outdoor conditions was similar to that of a corridor. The TSN and P2P connection were established within 35-40 meters range. The P2P message exchange could be done within the range of 90-100 meters.

Chapter 6

Conclusion and Future work

We modified the AlertApp to implement Readers-Writers Flow Model (RWFM). We successfully achieved message privacy as we can send any message only to the intended users by adding them to our reader's list. We are also following rules of updating the labels as defined in RWFM. Transient Social Network can be created from within the application. Peer to peer messaging in case of Internet failure has also been achieved. The emoticon representation of disaster conditions is achieved.

Currently, Data 61 team is working on the integration of their disaster management application with social networks, like Twitter handles. There is a scope of work in the integration of social networks with the AlertApp (which is a part of the disaster management application), to receive the emergency alerts directly. Further mobile registered can be verified by OTP confirmation to prevent Sybil attack on the application.

The major drawback of the wifi is the range. During an event of a disaster, it is really important to get the message across at longer distance. One method we propose is to use multi-hop message. This multihop technique uses P2P message at the base. Each device receives a P2P message from other device and communicates this message along with its own message to a third device. Using this technique, the third device will have message and

location of the first and second device. This third device, using P2P message, can further communicate the message it has received from the second device to a fourth device, and so on. The message will contain a sender's list which will have an ID of all the devices which have taken part in communicating the message. By this technique, the message chain will grow, and the range of message communication can be increased. We need to ensure that there is no loop formed by this technique. This can be done easily since we are maintaining the list of senders along with the message. Hence a device will not communicate a message to another device if the receiving device id is already on the sender's list of the message. This will prevent a loop from being formed.

The communication between different components needs security. This needs a lightweight encryption. Attribute based encryption [17] is a promising technique which reduces number of keys in broadcast scenarios. Ciphertexts are labeled with sets of attributes and private keys are associated with access structures that control which ciphertexts a user is able to decrypt.

In parallel work at IIT Kanpur by **Chigullapally Sriharsha** under **Prof R K Ghosh**, research was done to send the messages over disconnected network in case of disaster. The messages were saved in bundles and send over when the network became available. The local mule starts the TSN network and collects the messages, which can be send to other islands either when network is available or by passing on these messages to super mule who move between the islands.

This complements our work. Instead of having a single local mule, we have peer to peer messaging where we establish wifi links between any two users for the message exchange. This can be further expanded where each node bundles up the messages of the other node, just as the local mule does, and then pass it on to other nodes. Whenever any node connects to the Internet, these messages can be send over using the network. Each node in our case can double up as a local/super mule and can move to exchange and

bundle messages. This reduces dependency on single mule and at the same time, increases the range of connectivity where messages can be send.

We have further used the privacy model: RWFM to protect each node from possible breach of data. This can be extended to the work by Sriharsha wherein the messages can only be exchanged between the authenticated mules. The downgrade technique can be used by mules to exchange messages with the distress nodes. This is particularly important as any other node may masquerade as a mule. This can be prevented by specifying only mules as the readers in the reader's list of the distress node. Hence, this would prevent the possible privacy breaches during a case of disaster.

References

- [1] "2013 Kumbh Mela stampede". https://en.wikipedia.org/wiki/2013_Kumbh_Mela_stampede. Accessed: 2016-10-12. 1.1
- [2] "Disaster Management and Social Media - a case study". <https://www.police.qld.gov.au/corporatedocs/reportsPublications/other/Documents/QPSSocialMediaCaseStudy.pdf>. 1.1
- [3] Model. https://en.wikipedia.org/wiki/Biba_Model. Accessed: 2017-06-27. 3.1.2
- [4] Model. <https://developer.android.com/training/articles/user-data-ids.html>. Accessed: 2017-06-27. 5.3
- [5] "Steps of Disaster Management". <https://www.emaze.com/@AZFCRIZQ/disaster-management-copy1>. (document), 1.2
- [6] "Tsunami". <http://www.go-green.ae/The-Anatomy-of-a-Tsunami/117>. (document), 1.1
- [7] D. Bell and L. La Padula. "Secure computer systems: Unified exposition and multics interpretation". *Technical Report ESD-TR-75-306, MTR-2997, MITRE, Bedford, Mass*, 1976. 3.1.1
- [8] Matt Bishop. *"Introduction to Computer Security"*. Addison-Wesley, 2004. 3.1.1

- [9] Matt Bishop. "*Introduction to Computer Security*". Addison-Wesley, 2004. 3.1.1
- [10] D. E. Denning. "A lattice model of secure information flow". *Commun. ACM*, 19(5):236–243, 1976. 3.2
- [11] Miller, Engemann and Yager. "Disaster Planning and Management". *Communications of the IIMA, Volume 6 Issue 2*, 2006. (document), 1
- [12] A. Bhatnagar , A. Kumar, R. K. Ghosh, and R. K. Shyamasundar. "A Framework of Community Inspired Distributed Message Dissemination and Emergency Alert Response System over Smart Phones". *8th International Conference on Communication Systems and Networks (COM-SNETS)*, 2016. 4
- [13] K.Biba. "Integrity considerations for secure computer systems". *Technical Report ESD-TR-76-372, MITRE, Bedford, Mass*, 1975. 3.1.2
- [14] N. V. Narendra Kumar and R. K. Shyamasundar. "Realizing purpose-based privacy policies succinctly via information-flow labels". *IEEE 4th BdCloud*, pages 753–760, 2014. 1.1, 1.3, 3.3
- [15] N. V. Narendra Kumar and R. K. Shyamasundar. "POSTER: dynamic labelling for analyzing security protocols". *ACM 22nd CCS*, pages 1665–1667, 2015. 1.3, 3.3
- [16] Rajkumar Buyya, Raj Gaire, Ratan K Ghosh, Surya Nepal, Deepak Puthal, Rajiv Ranjan and Rudrapatna K Shyamasundar. "Cloud4BigData: A Scalable Cloud Service for Big Data Applications". Manuscript. (document), 1.3, 1.1, 1.4
- [17] Vipul Goyal, Omkant Pandey, Amit Sahai and Brent Waters. "Attribute-based encryption for fine-grained access control of encrypted

- data". *Proceedings of the 13th ACM conference on Computer and communications security*, pages 89–98, 2006. 6
- [18] Chigullapally Sriharsha and Prof. R.K. Ghosh. "An Opportunistic Network Architecture for Dissemination of Emergency Messages". *M. Tech. Thesis, IIT Kanpur*, 2017. (document), 2, 2.1
- [19] M. B. Sinai, N. Partush, S. Yadid, and E. Yahav. "Exploiting social navigation". *Black Hat Asia, CoRR:abs/1410.0151*, 2015. 1.2