Evaluation of Trust in Android Devices using Reinforcement Learning

Software Requirement Specification Document

Comprehensive Design & Analysis Project - 2016
B.Sc. Special (Honors) Degree in Information Technology

Project ID: 16J -008

Submitted Date: 9/15/ 2016

Batch: June Intake 2016

Author:

Student ID	Name	Signature
IT13067816	T.D.H Priyadarshani	

Supervisor

……………………………

Mr. Lakmal Rupasinghe

Co – Supervisor

……………………………

Mr. Krishnadeva Kesawan

Declaration

“We declare that the project would involve material prepared by the Group members and that it would not fully or partially incorporate any material prepared by other persons for a fee or free of charge or that it would include material previously submitted by a candidate for a Degree or Diploma in any other University or Institute of Higher Learning and that, to the best of our knowledge and belief, it would not incorporate any material previously published or written by another person in relation to another project except where the acknowledgement is made in the text.”

Student ID	Name	Signature
IT13067816	T.D.H Priyadarshani	

Table of Contents

1. Introduction

This section of the Software Requirement Specification (SRS) mainly provides a brief description and an entire overview of the overall system to be described. This section includes purpose, definitions, acronyms, abbreviations, references and overview of the SRS.

1.1 Purpose

The purpose of this document is to provide a detailed description and understanding of the Calculation of Android device Trust based on Malware Analysis using Reinforcement Learning Project. It will explain product's target audience, system constraints, interfaces and interactions with other external applications, features of the system, functional requirements, nonfunctional requirements, data requirements, quality requirements, hardware requirements and software requirements as well as Issues related. SRS will depicts how each component is interrelated and how they communicate with each other to bring the final result.

The ultimate purpose of this specific document is to give an idea to the audience of how User Level details are obtained using Social Media and how these extracted features of the user including other contributing features are combined into one feature vector and finally how Reinforcement learning is used to predict the Final trust value of the device.

1.2 Scope

This document is concerned about the feature extraction at user level which contributes to estimating the overall trust value of the device in MANETS. A Reinforcement Learning technique which is Q-learning will be used to determine the overall trust of a device.

Analyzing information of user plays an important role because we need to determine whether the user is a malicious user or not. When determining the overall trust of the device, aspects such as Operating System features, Network Layer information, Features of apps installed in the device are considered in addition to User Details. The Final Product is to gather all these features into a feature vector and carry out a malware analysis using Q-Learning and determine the overall trust of the device.

1.3 Definitions, Acronyms and Abbreviations

Definitions

Android Device – A Mobile device which uses Android as its Operating System

Android - Java and XML supported language used to develop software for android based mobile devices.

Acronyms

SRS	Software Requirements Specification
SNS	Social Network Site
FB	Facebook
RL	Reinforcement Learning
OS	Operating System
MANETS	Mobile Ad-hoc Networks

1.4 Overview

This research is based on calculating the trust of a device based on multiple features by doing a Malware Analysis. When determining the trust of the device many aspects should be taken into consideration. But since machine learning algorithm is used to predict the overall trust based on features, the features we can use without degrading the performance is limited [1]. Therefore we mainly focus on four main aspects which is User, Network, Applications installed in the device, Operating System. We take features from each of the four aspects and combine it to form a feature vector. This feature vector is used to predict the final trust of the device using a Reinforcement Learning Algorithm.

MANETS have various types of nodes with different characteristics such as malicious nodes, selfish nodes and etc. The behavior of a node and which type of node the device is examined using the collected features. We found that malware contribute a high percentage to degrade the overall trust of the device [2]. Therefore by using applications installed we are looking for malicious applications in the device. By examining the user we are checking whether the user is malicious user or not and etc.

The main goal of this research is to determine the overall trust of the devices in MANETS and make a secure network by connecting the devices with high trust values for effective and secure communication between two nodes. The ultimate aim is to provide a secure way to send sensitive data among devices in MANETS.

This document is divided into two sections. The main purpose of this document is about Information Extraction of the User or the owner of the device and how it contributes to the overall trust value. This document contains constraints, dependencies, assumptions and other important factors that affect the development and design of this project. The first section describes the functionalities and behavior of the system. The second section focuses on non-functional requirements and other requirements of the system. Use case diagrams and other UML diagrams are used to give a clearer idea about the system.

The latter part of the document contains the appendices and the References. These can be used to obtain further information such as appropriate algorithms and equations that will be used in the development of this process and prove how we derived certain facts and information about the project.

There are three things that should be carried out during user information gathering

1. Analyze the SNS (which is Facebook in this context) and gather required information.
2. Feature Extraction of Gathered Information.
3. Send the obtained user details to make the final feature vector with other features which is ultimately used to calculate the overall trust of the device.

2.0 Overall Descriptions

MANETS are a type of network structure which is composed of mobile devices. They are ever changing, ad-hoc and infrastructure less. MANETS rely on nodes which are mobile devices to forward packets to and from each other even if the packets don't belong to them. This is vital to the performance of the MANETS. Constant forwarding of packets may degrade the performance and use up resources of mobile devices. Therefore some of these devices may be reluctant to forward these packets. These nodes are known as selfish nodes. Since MANETS contain a diversity of mobile devices some of these devices could be malicious devices or may contain malware without their knowledge. These nodes are referred to as malicious nodes. All these types of unhealthy nodes will degrade the performance of the MANETS. If a device wants to send some important or sensitive data to another device through a MANET we cannot be sure whether this information will be protected and may not reach the wrong hands or whether we the data has been forwarded to the correct destination. This is where device trust plays an important role. Therefore before sending some date trust of the devices that encounter in the path should be known so that sensitive data will not reach wrong hands.

Trust of a device can be looked at in various aspects. In this context we are looking at trust in 4 angles.

1. Applications Installed
2. Operating System
3. Network Information
4. User/Owner of the device

Features from all 4 of these aspects will be collected and used to form the dataset. A model will be created using a Reinforcement Learning Algorithm and it will be used to predict the overall trust. We are planning to use Q-Learning Reinforcement Learning technique to this process.

Features that affect the trust at user level will be discussed in this section. The parameters that are taken into account are

1. Number of followers
2. Number of friends
3. Followers to Friends Ratio
4. Designation of the user
5. Density of interactions of the user
6. Location of the user

There are three things that should be carried out during user information gathering

1. Analyze the SNS (which is Facebook in this context) and gather required information.
2. Feature Extraction of Gathered Information.
3. Send the obtained user details to make the final feature vector with other features which is ultimately used to calculate the overall trust of the device.

   Reinforcement Learning is a widely used machine learning technique which is created based on how human brain works and psychology. It always tries to choose the optimal answer. By using this technique we are planning to improve the accuracy of our trust Evaluation.

2.1 Product Perspective

Numerous studies were carried out to determine the device trust of Android devices considering various aspects. Most of them are concerned on MANETS and peer-to-peer networks. And there are limited numbers of studies carried out on Device trust for IoT. Some of the studies are determine device trust based on Malware Analysis and some of them are based on other factors such as User's information.

A study carried out by Weiss and Reznik [5] on Trust Evaluation in Mobile Devices stated that Trust evaluation should integrate various metrics ranging from accuracy and reliability of the data sources to the security of the procedures and tools used. In order to fill these criteria they used application details, device feature security, sensor security, Battery Usage, CPU Usage, Network usage and level of privacy provided by the device as parameters. All these parameters are considered for the identification of malicious behavior. Since malicious program tend to use up more resources this method is effective. Yet they have failed to capture the user's details which represent trust just as much as other factors.

Zhao and Pan [2] introduced a Machine Learning based Trust Evaluation Framework for Online Social Networks. They have considered numerous factors about users such as the density of interactions between the two users, the similarities between the two users, the number of mutual friends between the two users and etc. Although this is a good approach they only give the final output as two values, i.e. whether we could trust or distrust a particular device. This approach isn't ideal to be used in an IoT environment or MANETS because we cannot completely disregard all the untrusted devices. It will degrade the performance. So rather than completely saying a particular user or device is untrusted, assigning a continuous value as the final output is more applicable cause then it allows us to compare.

Bao and Chen [4] introduced a Dynamic Trust Management for the IoT Applications. They provided a flexible and accurate trust assessment for IoT entities. Although their trust assessment is quite accurate and successful they have failed to detect the presence of malicious behavior and selfish nodes. They have mainly focused on user's relationships.

Trust or Reputation analysis is looked upon in various angles. Presence of malicious programs affects trust of the device most because most of the time the user is unaware about the presence of malware. We believe that presence of malware can be analyzed and confirmed in various aspects. Various studies are based on malware detection and analysis. There are static analysis techniques, dynamic analysis techniques as well as hybrid approaches. There are multiple ways how malware get installed into devices and several different ways they get activated. They could exploit the vulnerabilities in the user level, network level, operating system and applications installed in the device. Therefore it is necessary to identify the weakness at each of these levels because all of these weaknesses will diminish the trust of the devices and provide many opportunities for malware to get installed in our devices.

So far there are large numbers of Android Malwares roaming around. ''Among all mobile malware, the share of Android based malware is higher than 46% and still growing rapidly." [6] These malware are diverse and how they get activated and the consequences they cause and where they cause these consequences are varied. Therefore in order to identify all of these malware present in a device, looking at only one aspect won't work.

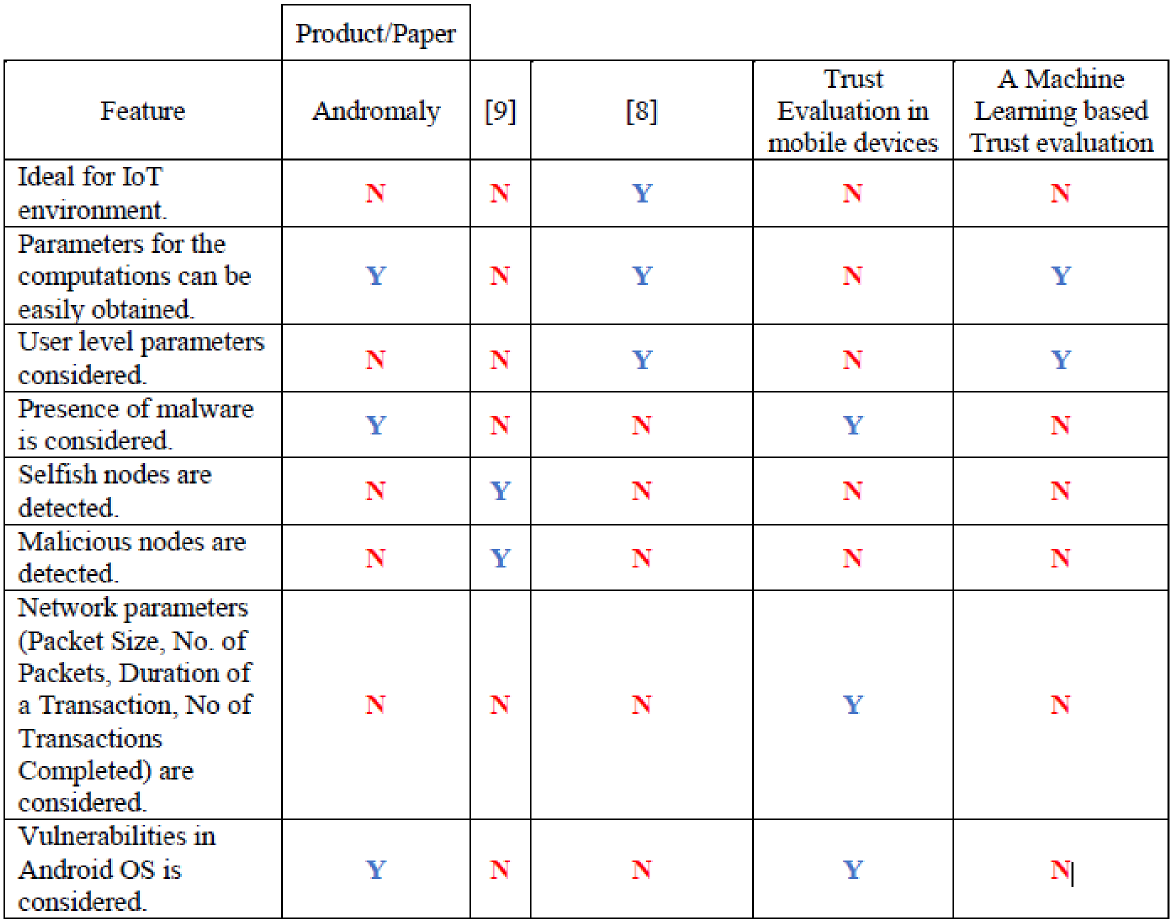
Weaknesses Malware uses:

- Using System events and System calls
- Vulnerabilities in the OS such as bugs, configuration oversight, designation
- Vulnerabilities in the applications installed such as bugs, configuration oversight, designation
- Some Malicious Payloads cause Remote Control. This can be analyzed using network information [6].
- Some Malicious Payloads cause Financial Charges, by subscribing the user to premium rate services without user's awareness. These things happen in the background. Therefore it is necessary to detect how much data or resources a particular app takes up in the background.

Yerima and Sezer [7] introduced a Malware Detection Mechanism using Parallel Machine Learning classifiers. As the features for the feature vector they extracted API related features, App permissions and Standard OS and Android framework commands. Although they have considered both application and OS related information to detect the malware they have neglected the network and user related information which is just as important.

Mekouar, Iraqi and Boutaba[8] brought forward a method called Detecting Malicious Peers in a Reputation-Based Peer-to-Peer System. They allocate the reputation based on the feedback and Authenticity of the files given. Although this is a good approach to be used in a peer-to-peer system we cannot use this in an IoT environment cause it's not practical to collect feedback.

Our end product focus on deriving the trust of the device as mentioned above by looking at the presence of malware and other important aspects. So far no approach has been used to Evaluate Trust using Reinforcement Learning technique according to our knowledge. Most of the current procedures and products have determined trust using only one aspect. But Device trust is something that should be looked at in various angles. Our approach looks at trust in 4 different aspects and can be used to determine the true trust value of a device and provide a secure way to send sensitive data through MANETS.



	Product/Paper				
Feature	Andromaly	[9]	[8]	Trust Evaluation in mobile devices	A Machine Learning based Trust evaluation
Ideal for IoT environment.	N	N	Y	N	N
Parameters for the computations can be easily obtained.	Y	N	Y	N	Y
User level parameters considered.	N	N	Y	N	Y
Presence of malware is considered.	Y	N	N	Y	N
Selfish nodes are detected.	N	Y	N	N	N
Malicious nodes are detected.	N	Y	N	N	N
Network parameters (Packet Size, No. of Packets, Duration of a Transaction, No of Transactions Completed) are considered.	N	N	N	Y	N
Vulnerabilities in Android OS is considered.	Y	N	N	Y	N

2.1.1 Hardware Interfaces

- Mobile Phone

This is the main and important hardware device involved in the project as trust is determined for each mobile device in a MANET. The mobile devices have various manufacturers, brands and Operating systems. In this project a mobile device involved should be an android based smart phone.

- Server

Although Mobile Devices provide a lot of facilities they are not yet capable of running and training machine learning algorithms. For this purpose server with sound processing power should be used to calculate the trust of each mobile device by running the Reinforcement Learning Algorithm.

2.1.2 Software Interfaces

- Android Operating System

There are various OS used in mobile devices but for this context the mobile device used should have Android OS installed since this is based on Android OS and applications it supports. The mobile device should have Wi-Fi capabilities and should have the facility to connect to Facebook.

- Web Server

A server should be used to keep the model and calculate the trust value for the mobile devices in the network. It should be able to request necessary information from the Mobile devices and send them back their computed trust value.

2.1.3 Communication Interfaces

3G **-** 3G connection of the mobile phone will be used for data transmission between the mobile device and the server. Wi-Fi **-** If the mobile data is not available, user can connect to an available Wi-Fi router to get the internet connection in order to transfer data to and fro. This will be used for data transmission between the mobile devices and the server.
Required Connection bandwidth might differ time to time. Since large data load from different mobile devices are travelling through the network, having a high bandwidth internet connection will be useful for the performance of the MANETS.

2.1.4 Memory Constraints

Server Memory – since the server has to process data from several mobile devices large memory will be required.

Mobile Device Memory - Minimum of 1GB memory will be sufficient

2.1.5 Operations

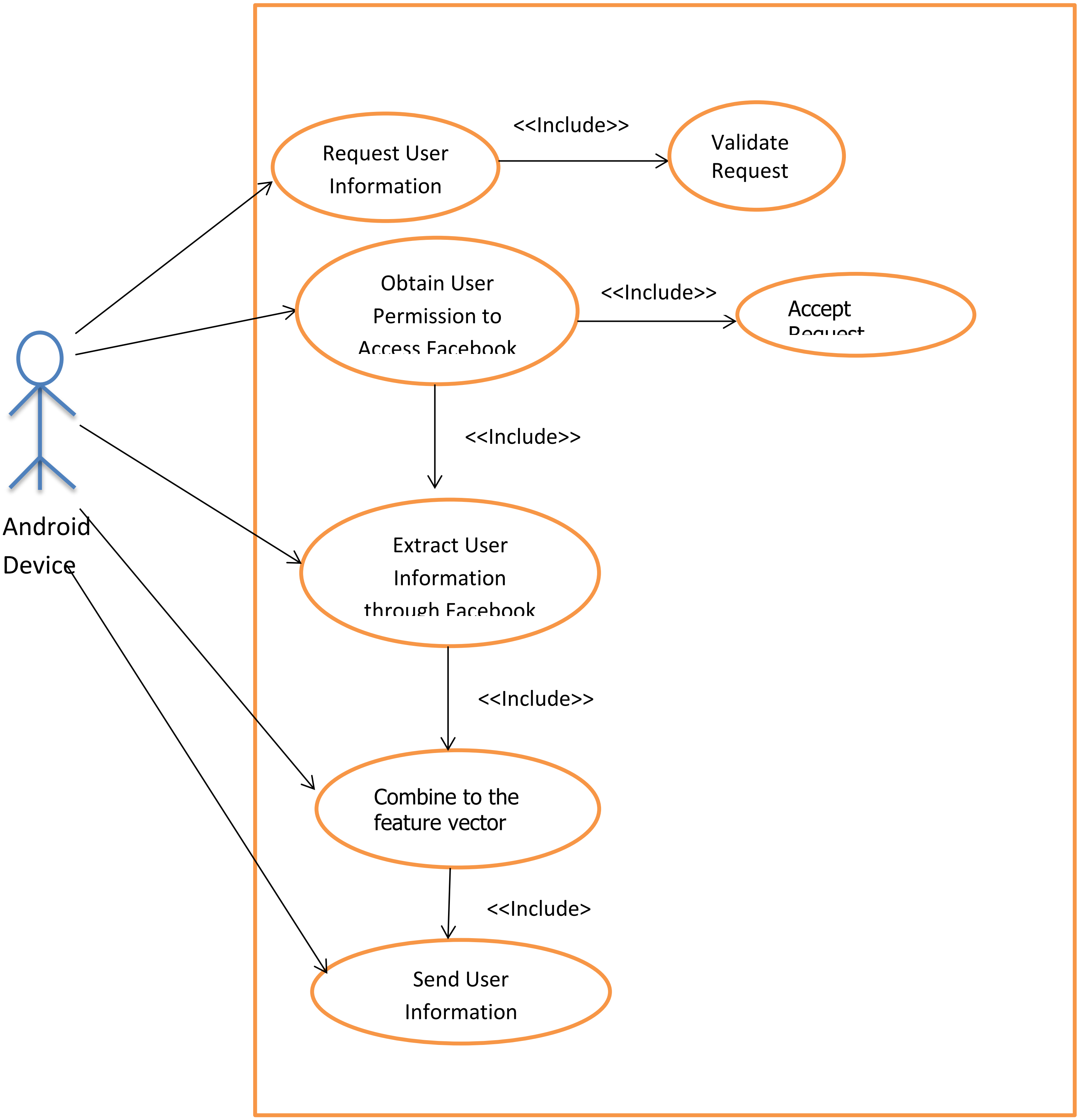
The Mobile Device user is capable of accepting or declining requests to obtain certain details from the mobile device such as personal information, other than that there will be very less user involvement.

2.2 Product Functions

1. Data Collection
    - A dataset with the required attributes should be collected to train and build the model.
2. Model Building and Training using Reinforcement Learning
    - Train the model Using the dataset
    - Build the model
    - Use the model for new predictions
    -

3. Predict New Values
    - Obtain the feature vector from the mobile device
    - Sent the parameters of the feature vector through the model
    - Obtain the Estimated Trust Value
    - Assign the trust value to the mobile device

- Use Case Diagram



- Use Case Scenarios

Use Case Name	Request User Information
Preconditions	The Devices should be connected by a network.
Successful End Condition	An Interface requesting User Permission
Actor	Android Mobile Device
Main Success Scenario	1. Request Received 2. Check whether the Request is Authentic. 3. Display an Interface
Extensions	If the request is not authentic, it will be ignored.

Use Case Name	Obtain User Permission to Access Facebook
Preconditions	1. The request should be Authentic 2. The user should have a Facebook account. 3. The User should be connected to Facebook through the android Device
Successful End Condition	Access to Obtain Information
Actor	Android Mobile Device
Main Success Scenario	1. An interface Requesting User Permission 2. User Permits to Access Facebook and retrieve required Information.
Extensions	If the User is not using Facebook or not connected to Facebook through the android device the Permission will not be granted.

Use Case Name	Extract User Information through Facebook
Preconditions	1. The required information should be filled. 2. The information provided by the user in Facebook should be Authentic.
Successful End Condition	Required Information Extracted from Facebook
Actor	Android Mobile Device
Main Success Scenario	1. Access Facebook 2. Search for required Information 3. Retrieve Required Information
Extensions	If the Details are not provided by the user, required information will not be retrieved.

Use Case Name	Send User Information
Preconditions	The Devices should be connected by a network. The sender should be aware of the destination location.
Successful End Condition	The Extracted User Information sent to the server
Actor	Mobile Device
Main Success Scenario	1. Connect with the Server 2. Send the extracted information successfully. 3. Receive an Acknowledgement.
Extensions	If the two devices are not connected by a network properly the extracted information will not be sent.

2.3 User Characteristics

This approach mainly target users that use MANETS and require to send their sensitive data in a secure manner.

2.4 Constraints

2.4.1 Hardware Limitations

To use this service, user should have an android smart phone with wifi and network capabilities. The smart phone should have connectivity to the internet and access to fb.

3. CPU : Dual-core 1.5 GHz
4. 3G
5. RAM : 1 GB

2.4.2 Software Limitations

This service will require Android Jelly Bean (4.2) or upwards.

2.5 Assumptions and Dependencies

When we are developing this service we assume:

- Most users have Android mobile devices.
- Users who interact with the system have at least a slight knowledge about handling smart phones.
- All users have a good Internet connectivity.
- The bandwidth of the Internet connection will not effect to the data transfer.
- All the users use Facebook and they are connected to their profiles using smartphones.
- Server has sufficient processing power to handle all the requests without a noticeable delay.
- The information provided by the user to social media is true.

2.6 Apportioning of Requirements

The requirements described in sections 1 and 2 of this document are referred to as primary specifications; those in section 3 are referred to as requirements (or functional) specifications. The two levels of requirements are intended to be consistent. Inconsistencies are to be logged as defects. In the event that a requirement is stated within both primary and functional specifications, the service will be built from functional specification since it is more detailed.

Essential requirements (referred to in section 3) are to be implemented for this version of Project. Desirable requirements are to be implemented in this release if possible, but are not committed to by the developers. It is anticipated that they will be part of future release. Optional requirements will be implemented at the discretion of developers.

3 Specific Requirements

3.1 External Interface Requirements

3.1.1 Hardware Interfaces

- Mobile Phone

This is the main and important hardware device involved in the project as trust is determined for each mobile device in a MANET. The mobile devices have various manufacturers, brands and Operating systems. In this project a mobile device involved should be an android based smart phone.

- Server

Although Mobile Devices provide a lot of facilities they are not yet capable of running and training machine learning algorithms. For this purpose server with sound processing power should be used to calculate the trust of each mobile device by running the Reinforcement Learning Algorithm.

3.1.2 Software Interfaces

4. Android Operating System

There are various OS used in mobile devices but for this context the mobile device used should have Android OS installed since this is based on Android OS and applications it supports. The mobile device should have Wi-Fi capabilities and should have the facility to connect to Facebook.

5. Web Server

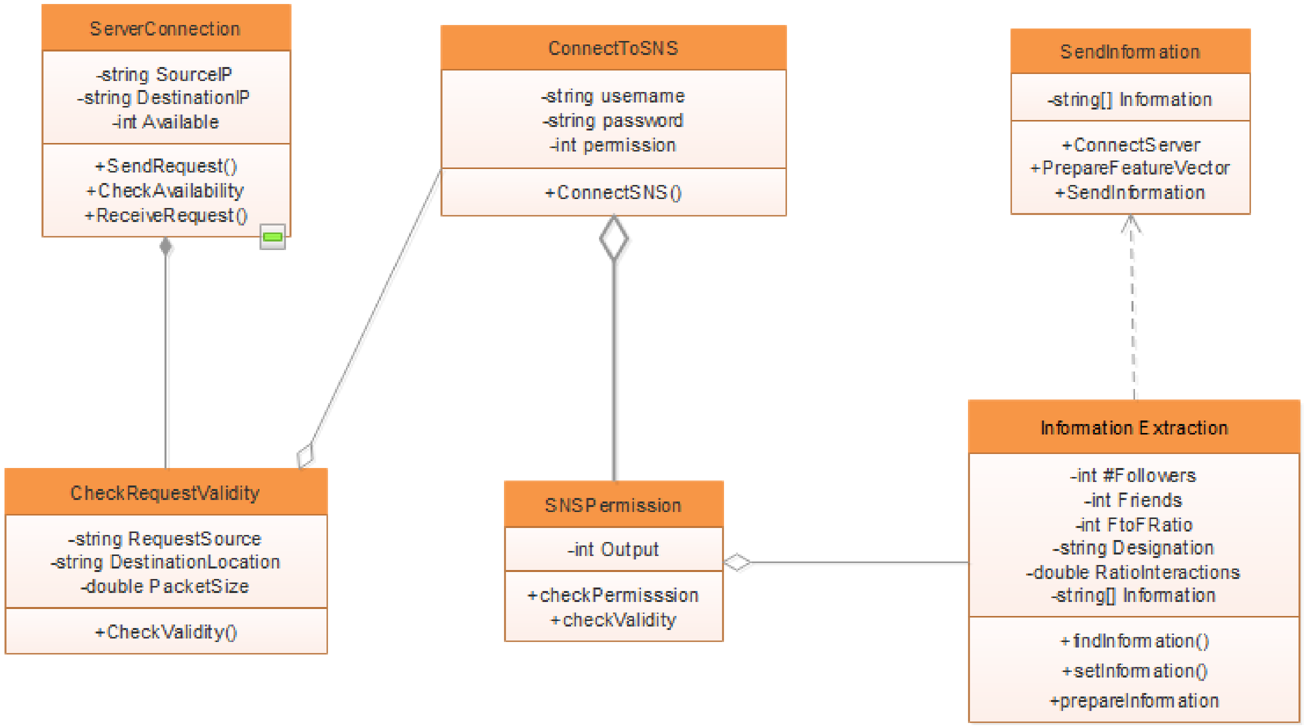
A server should be used to keep the model and calculate the trust value for the mobile devices in the network. It should be able to request necessary information from the Mobile devices and send them back their computed trust value.

3.1.3 Communication Interfaces

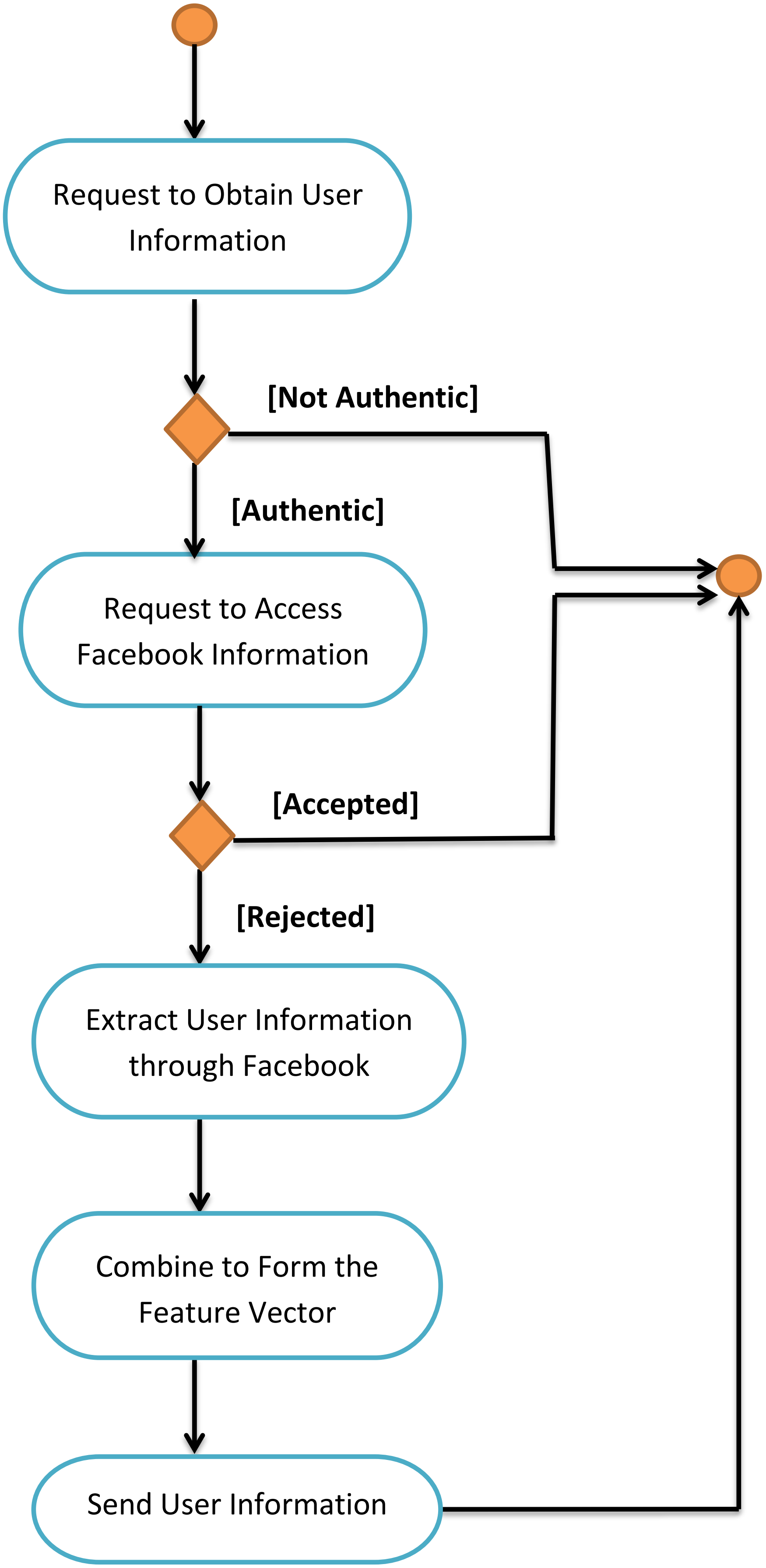
- 3G **-** 3G connection of the mobile phone will be used for data transmission between the mobile device and the server.

- Wi-Fi **-** If the mobile data is not available, user can connect to an available Wi-Fi router to get the internet connection in order to transfer data to and fro. This will be used for data transmission between the mobile devices and the server.
- Required Connection bandwidth might differ time to time. Since large data load from different mobile devices are travelling through the network, having a high bandwidth internet connection will be useful for the performance of the MANETS.

3.2 Class/Objects



3.3 Activity Diagram for User Information Extraction


Request to Obtain User Information
[Not Authentic]
[Authentic]
Request to Access Facebook Information
[Accepted]
[Rejected]
Extract User Information through Facebook
Combine to Form the Feature Vector
Send User Information

3.4 Performance Requirements

- Supporting Instances - System will allow only one instance of this service to run on the mobile device at a given time.

- Simultaneous Users – the application will allow multiple devices to use this service but not from the same mobile device. With several devices, many users can access the server and obtain services.

3.4.2 Design Constraints

System will use standard principles of designing and it will maintain the consistency. There will be no specific design constraints to the project.

3.5 Software System Attributes

3.5.1 Reliability

This service has a chance of failing due to OS failures such as low battery or system crashes. But since the server is handling most of the computation part results will be accurate almost all the time.

3.5.2 Availability

As long as the server is not down or there is some sort of network problem which will cause devices to reach the server difficult and vice versa the availability of the system will be high.

3.5.1 Security

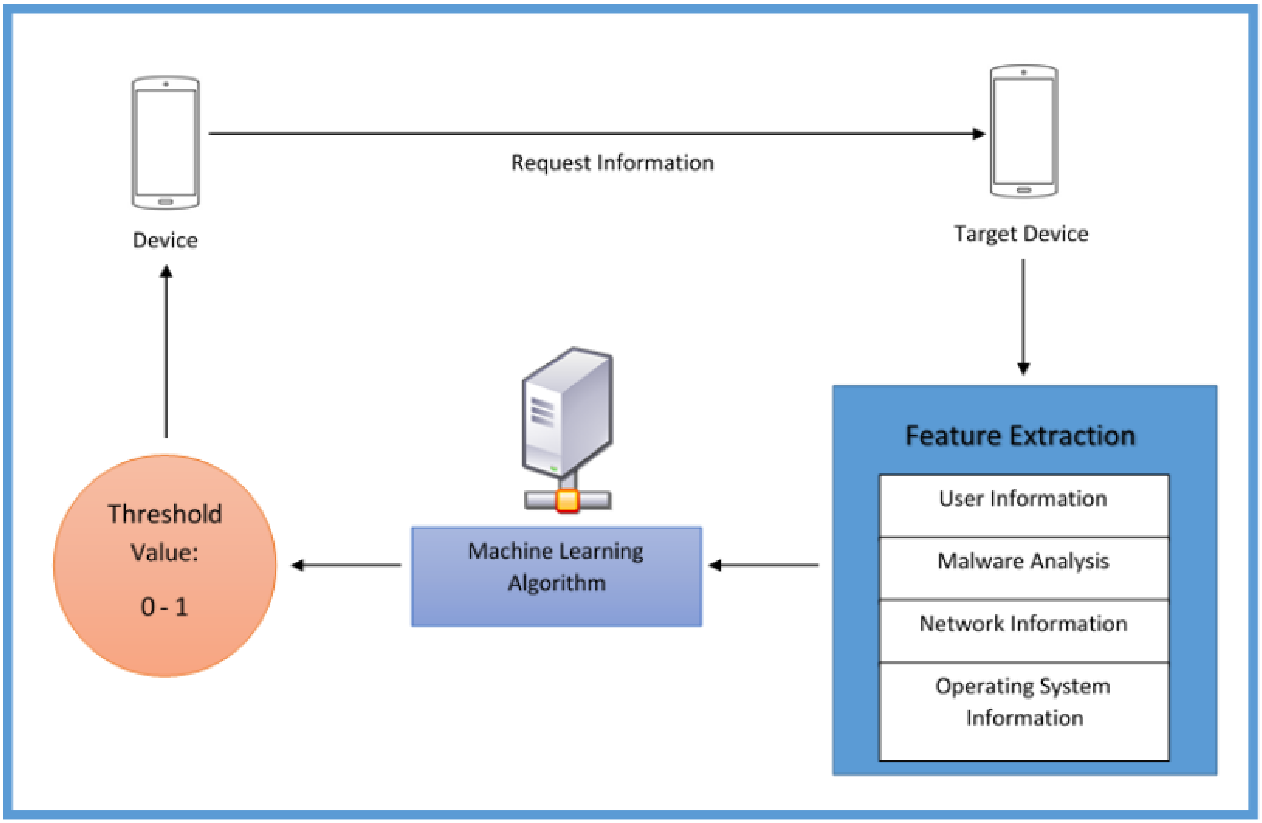
The messages that will be sent to and from server to the devices will be encrypted. And information of the mobile devices and their owners will not be obtained without their consent.

3.5.3 Maintainability

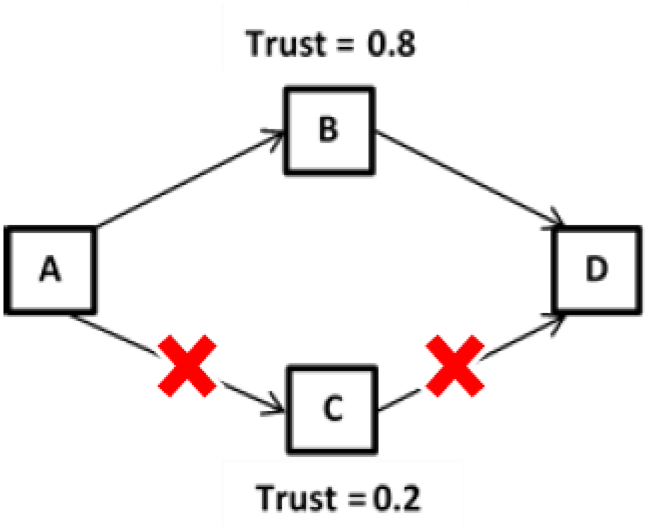
The service will be installed to android devices as an Android Service. As most of the logic and computation is in the server, Server crashes and failures should be fixed.

4. Supporting Information

4.1 System Diagram



4.2 Other Diagrams



The above diagram demonstrate how calculating trust ultimately help to build a secure network that can be used to transfer sensitive information.

5. References

1. [1] K. Rose, "Internet Society Releases Internet of Things (IoT) Overview Whitepaper: Understanding the Issues and Challenges of a More Connected World | Internet Society", Internetsociety.org, 2015. [Online]. Available: https://www.internetsociety.org/blog/public-policy/2015/10/internet-society-releases-internet-things-iot-overview-whitepaper. [Accessed: 30- Jul- 2016].
2. [2] "IDC: Smartphone OS Market Share", www.idc.com, 2016. [Online]. Available: http://www.idc.com/prodserv/smartphone-os-market-share.jsp. [Accessed: 31- Jul- 2016].
3. [3]"G DATA Releases Mobile Malware Report for the First Quarter of 2015", G DATA Software AG, 2016. [Online]. Available: https://www.gdata-software.com/g-data/newsroom/news/article/g-data-releases-mobile-malware-report-for-the-first-quarter-of-2015. [Accessed: 31- Jul- 2016].
4. [4] 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, 2012.
5. [5] 2016. [Online]. Available: https://www.quora.com/Which-tools-are-available-to-collect-social-network-data. [Accessed: 03- Aug- 2016].
6. [6]"Social Network Analysis", Slideshare.net, 2016. [Online]. Available: http://www.slideshare.net/gcheliotis/social-network-analysis-3273045. [Accessed: 03- Aug- 2016].
7. [7] D. Desale, "Top 30 Social Network Analysis and Visualization Tools", Kdnuggets.com, 2016. [Online]. Available: http://www.kdnuggets.com/2015/06/top-30-social-network-analysis-visualization-tools.html. [Accessed: 03- Aug- 2016].
8. [8] I. Chen, F. Bao, M. Chang and J. Cho, "Dynamic Trust Management for Delay Tolerant Networks and Its Application to Secure Routing", IEEE Trans. Parallel Distrib. Syst., vol. 25, no. 5, pp. 1200-1210, 2014.
9. [9] L. Mekouar, Y. Iraqi and R. Boutaba, "Peer-to-peer’s most wanted: Malicious peers", Computer Networks, vol. 50, no. 4, pp. 545-562, 2006.