3-2018

Data Aggregation Technique to Provide Security for Wireless Sensor Networks

Anusha C. Thottempudi
St. Cloud State University, anushachowdhary94@gmail.com

Data Aggregation Technique to Provide Security for Wireless Sensor Networks

by

Anusha Chowdhary Thottempudi

A Starred Paper

Submitted to the Graduate Faculty of

St. Cloud State University

in Partial Fulfillment of the Requirements

for the Degree of

Master of Information Assurance

March, 2018

Thesis/Starred Paper Committee:
Susantha Herath, Chairperson
Lynn Collen
Balasubramanian Kasi

Abstract

Due to restricted computational power and energy assets, the aggregation of information from numerous sensor nodes is performed at the aggregating node and is typically done by using basic techniques, for example by averaging. Node compromising attacks more likely occur after such sort of aggregations of data. As wireless sensor networks are generally unattended and do not use any tamper resistant equipment, they are extremely vulnerable to compromising attacks. Therefore, determining the trustworthiness of information and the reputation of sensor hubs is vital for wireless sensor networks. As the execution of low power processors drastically enhances, future aggregator nodes will be equipped for performing more refined information aggregation algorithms, in this way making WSN less vulnerable. WSN stands for Wireless Sensor Networks. For this reason, Iterative algorithms hold high value. These algorithms take the data aggregated from different sources and give a trust appraisal of these sources, generally in the form of comparing weight variables which are given to information obtained from every source. In this paper, we show that few existing iterative filtering calculations, while altogether more vigorous against collusion attacks than the basic averaging methods, are in fact susceptive to a novel refined collusion attack which we launch. To address this security issue, we propose a change for iterative filtering procedures by giving an underlying estimation to such algorithms which make them collusion resistant as well as more precise and faster for merging purposes.

Acknowledgements

I would like to thank all the committee members Dr. Herath, Collen and Dr. Kasi for their time, encouragement and valuable suggestions. This paper would not have been possible without their guidance.

Table of Contents

Chapter Page

List of Tables

Table Page

List of Figures

Chapter 1: Introduction

Due to the requirement for observing the robustness and minimal cost of the nodes, wireless sensor system networks (WSN's) are normally repetitive. Information from various sensors is accumulated at an aggregator node which then advances the data to the base station just the total aggregate values. At present, because of impediments of the computing power and energy assets of sensor nodes, information is aggregated to a great degree with the help of basic calculations, such as for example, averaging of data. In any case, such aggregation is known to be extremely vulnerable against faults, and even more importantly against malevolent attacks (Ozdemir & Xiao, 2009). Cryptographic strategies can't prevent this, since the attackers for the most part gain full access to data present in the compromised nodes. Therefore, data aggregation at the aggregator node must be joined by an appraisal of the reliability of information from individual sensor nodes. In this way, more effective and modern algorithms are required for data aggregation in the future wireless sensor networks.

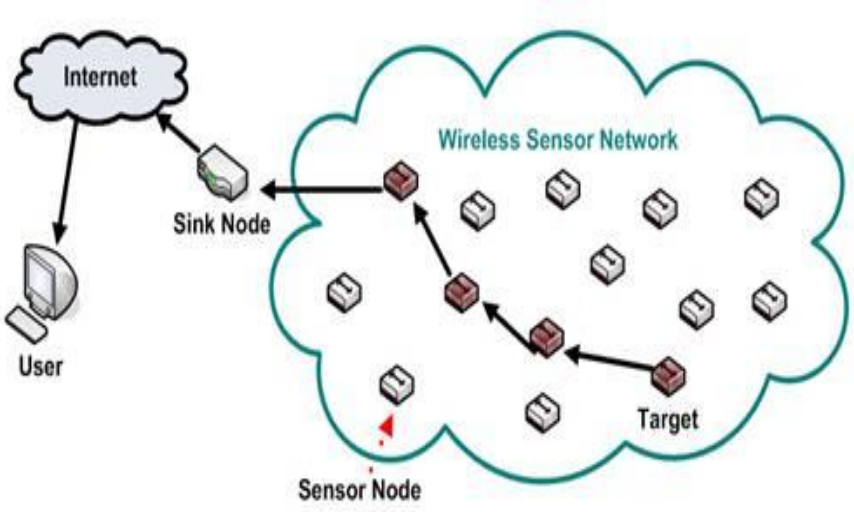

Figure 1.1 Wireless Sensor Networks (Shanika, 2015)

The required algorithms should at the very least have two important features: The first is non-stochastic errors like faults or any malicious attacks, those algorithms should provide data which is received from each sensor node which contains the assessment information related to the reliability and trustworthiness of data. And second of any random errors those algorithms should produce are estimates, which will be close enough to the optimal scenario.

Distributed systems, including e-commerce to most popular social networks use trust, reputation systems to assess their participants. They play an important role in the operation of different operating systems. At any particular time, the assessment of trustworthiness shows the participants aggregate behavior up to that particular moment. In a distributed system, attackers have many number of ways to modify the participant's trust and reputation values. Among the various number of ways this can be accomplished, aggregation algorithms are the primary target for the attackers.

Recent studies suggested that trust and reputation are very effective mechanisms to provide security in wireless sensor networks. In the case of distributed sensors, it is a challenging issue to assess the trustworthiness even though in many application domains sensor networks are used. Adversaries who plan to inject malicious data may perform node compromising attacks on sensors which are deployed in certain unfavorable environments. Due to the increase in computing power of processors and the simultaneous decrease in technology cost, it will be possible to implement highly complex algorithms with wireless sensor networks can afford highly configured hardware.

With the help of a single iterative procedure (Hoffman et al., 2009), the problems of both aggregation and assessment of data trustworthiness can be solved. This makes the usage of Iterative Filtering algorithms(IF) an attractive choice. For every sensor, the trustworthiness is calculated depending on the distance of sensor readings from the correct value estimates. The values are gathered from the readings of all sensors in the previous iteration round through some sort of aggregation method. This type of aggregation method is typically a weighted average.

Less trustworthiness is assigned to sensors for which the readings differ significantly from the estimates. So, their readings in the current iteration round are given lower weights.

Problem Statement

Trust and notoriety have been recently recommended as a viable security mechanism for Wireless Sensor Networks. Although sensor systems are being progressively utilized in numerous application areas, evaluating the trustworthiness of the revealed information from these distributed sensors has remained a problematic and challenging issue.

Nature and Significance of the Problem

During recent times, there has been an expanding amount of research on IF algorithms for trust and notoriety frameworks (Kerchove and Dooren, 2016), (Ayday et. Al., 2009). The execution of IF algorithms within the sight of various sorts of faults and simply false information injection attacks have been studied, for instance in (Chou et.al., 2013) where it was applied to compressive detecting information in WSNs. In the past

literature, it was found that these calculations display better robustness contrasted with the basic averaging techniques. In any case, the past research did not consider more advanced collusion attack scenarios. If by chance that the attackers have more stateful of knowledge about the aggregation and its parameters, they can direct attacks on WSNs by misusing false information injection through various compromised nodes.

This paper presents an advanced intrigue collusion attack against various existing IF algorithms considering the false information injection attack. In such an attack situation, colluders endeavor to skew the total values by driving such IF calculations to converge to the skewed total values given by one of the attackers.

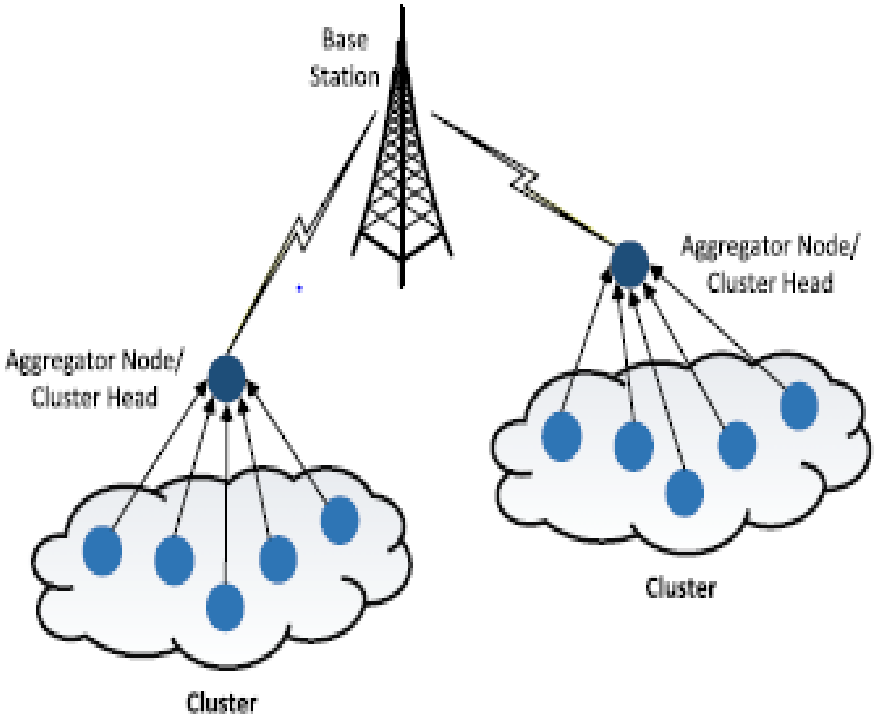

Figure 1.2. Network Model for WSN (Mohsen et al., 2014)

Objective of the Study

The objective of this study is to develop a data aggregation technique to provide security for wireless sensor networks.

Study Questions/Hypotheses

1. Why are Iterative Filtering (IF) algorithms used in Wireless Sensor Networks(WSN)?
2. How are collusion attacks launched against Wireless Sensor Networks dangerous?

Limitations of the Study

This study is not conducted on real time wireless senor networks such as sensor lights or smoke sensors. Few sensor nodes built in a computer application are considered as a network.

Definition of Terms

Wireless Sensor Network - Remote sensor systems (WSN) are spatially disseminated independent sensors to screen physical or ecological conditions. For example, temperature, sound and weight, and so forth and to agreeably go their information can go through the system to a principle area of collection. The advancement of remote sensor systems was inspired and driven by military applications. Today such systems are utilized as a part of numerous mechanical, commercial applications. For example, modern process checking and controls for machine health monitoring.

A Wireless Sensor Network is a type of remote network which incorporates a substantial number of circulating, self-coordinated, minute, low controlled gadgets named sensor hubs or motes. Mote is a wireless transceiver which can act as both sender and receiver. These networks unquestionably cover an immense number of spatially circulated, small, battery operated and embedded device. These are used to collect and carefully gather, process, and exchange information to the administrators, which has control over the capabilities of computing and handling. Nodes are the minor PCs, which work together to frame the systems.

The sensor node is a multi-practical device which is energy effective. The utilizations of motes in the field of industries are now widespread. An accumulation of sensor nodes gathers the information from the surroundings to accomplish particular application goals. The correspondence between motes should be possible with each other by utilizing single or multi line handsets. In a remote sensor arrangement, the quantity of bits can be at the request of hundreds or even thousands per transmission. With sensor networks that have Ad Hoc systems will have fewer nodes with no structure.

Wireless Sensor Network Architecture. The most widely recognized WSN follows the architecture model of the OSI design (Shanika, 2015). The design of the WSN incorporates five layers and three cross layers. Generally, in a sensor n/w we need five layers, specifically application layer, transport layer, network layer, data link layer and physical layer. The three cross planes are particularly in power administration, mobility administration, and task administration. These layers of the WSN are utilized to

achieve the network and make the sensors cooperate with a specific end goal to raise the total effectiveness of the network.

Application Layer. The application layer is subject to traffic administration and offers programs for various applications that change over the information in a reasonable timeframe to discover positive data. Sensor networks are organized in various applications and industries including agriculture, military, medical, and so on.

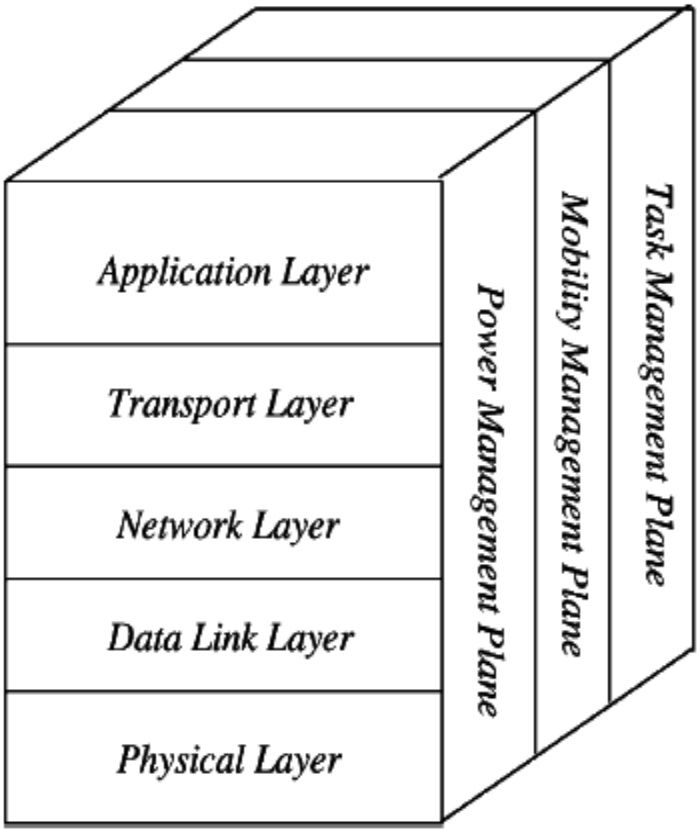

Figure 1.3. WSN Architecture

Transport Layer. The capacity of the transport layer is to provide traffic avoidance shirking and dependability where a considerable measure of conventions expected to offer this capacity and are useful on the upstream. These conventions utilize divergent components for recognition of loss and recovery of loss. The transport layer is precisely required when a framework is needed to contact different networks.

Giving a solid recovery of loss is more energy effective and that is one of the principle reasons why TCP is not a good fit for WSN. Usually, transport layers protocols can be isolated into either Packet driven or Event driven. There are some mainstream protocols in the transport layer, in particular, STCP which stands for (Sensor Transmission Control Protocol), PORT which stands for (Price-Oriented Reliable Transport Protocol and PSFQ which stands for (pump slow fetch quickly) which work better for WSN.

Network Layer. The principle requirement of the network layer is routing. It also has a lot of tasks in view of the application, including the fundamental tasks are in the power saving, incomplete memory, supports, and sensor doesn't have an ID which is universal and should be organized by itself.

The straightforward idea of the routing convention is to clarify a dependable path and repetitive paths, as indicated by a persuaded scale called a metric, which differs from convention to convention. There is a considerable measure of existing conventions for this network layer, they can be separated into; flat level routing, hierarchal routing or can be isolated into time driven, question driven and event driven.

Data Link Layer. The data link layer is at risk for multiplexing information frame recognition, information streams, MAC (Medium Access Control), error control and affirm the dependability of point–point (or) point– multipoint.

Physical Layer. The physical layer gives an edge to exchanging a flood of bits above physical medium. This layer is in charge of the frequency selection, carrier frequency generation, detection of the signal, modulation and information encryption. The IEEE 802.15.4 standard is recommended as run for low rate specific regions and remote sensor networks with ease of low cost, consumption of power, density, the scope of communication to enhance the battery life. There are a few different versions of IEEE 802.15.4.V.

Attributes of Wireless Sensor Network.

The attributes of WSN incorporate the following:

- Ability to deal with failures in the nodes.
- Some portability and heterogeneity for the nodes.
- Adaptability to substantial size of distribution
- Ability to guarantee strict natural conditions
- Easy to utilize
- Cross-layer plan
- Points of interest of Wireless Sensor Networks

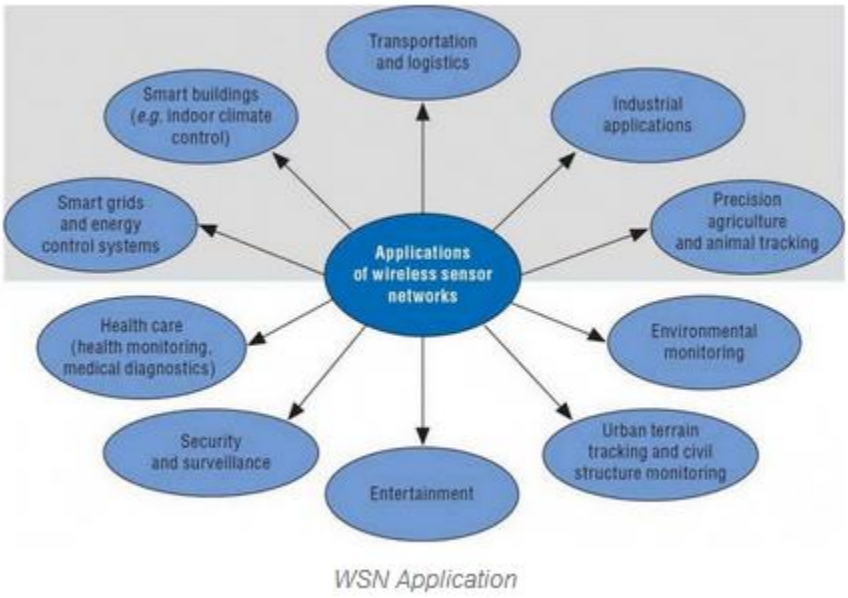

Figure 1.4. WSN Applications (Tarun, A, 2014)

The benefits of WSN incorporate the following

- Arrangements within the networks can be completed without moving the infrastructure.
- Adept for the difficult to reach spots like mountains, over the ocean, villages, and woods.
- Adaptable if there is a simple circumstance when an extra workstation is required.
- The price for the executions will be inexpensive
- It keeps away from a lot of wiring.
- It may give room to the new devices whenever it is necessary.
- A centralized monitoring can be used to open it.

Remote Sensor Network Applications. Wireless sensor systems may involve various distinctive sorts of sensors such as those with low rates of sampling or seismic sensors which are smart to screen an extensive variety of encompassing circumstances. Sensor nodes are utilized for consistent detecting, event ID, discovery, and nearby control of actuators. The uses of wireless sensors are incorporated into military, home, and other business uses.

Summary

In this chapter, wireless sensor networks are discussed along with the need to provide security for them. Also, the use of IF algorithms, their application and importance has been discussed. In the following chapter, an elaborate overview about wireless sensor networks and aggregation will be provided.

Chapter II: Background and Review of Literature

In this chapter background related to the IF algorithms and the network model and literature that helps to build a secure aggregation technique is discussed.

Background Related to the Problem

This section gives a brief overview regarding data communication, network classifications, topologies and types of networks. Transmission of digital information between at least two or even more computer systems or between a network of computers allowing the systems to exchange data is called data communication. Either wired cables or wireless remote media is used to establish connection between these devices. The Internet can be considered as the largest and best example for a computer network.

Computer networks can be defined as an arrangement of interconnected PC's (Personal Computers), servers and their electronic peripherals. For example, printer can be considered as a computerized peripheral. Data sharing between the computers is encouraged by this interconnection. Wired or remote wireless media can be used to connect these computers together into a network.

Classification of computer networks is done by taking into account basing on various different factors. They are as follows: geographical area, inter connectivity, administration and architecture.

Firstly, let us discuss the factor regarding the geographical span of a network. A geographical network can be any of the below categories:

1. Devices spread across a small or large table. For instance, Bluetooth device which span only a few meters.
2. Spread throughout the entire building, including transitional devices for connecting all of the floors together.
3. Spread throughout the entire city or across many cities.
4. A single network connecting the entire world.

Secondly, different ways of connectivity may be used to connect components of a network. Here, connectivity means it can be physical, logical or both. A network mesh can be used to connect every individual device to various other devices in a network. A bus structure can be used to connect all the devices to a single and geographically not connected medium. A linear structure can be used to connect every device to its peers which are both to the left and right of it. A star topology can be used to connect all the devices in the network to a single device. Devices which are connected in an arbitrary fashion by using all the above topologies can be called as hybrid topology network.

Thirdly, a network can be either private or public. A private network can be a single and autonomous which no person outside the network can access. A public network can be accessed by anyone.

Further, computer networks can be categorized into client server or peer to peer based on their architecture. A client makes a request to a server, and a response will be sent by the server depending on the request. The server may be one or many number of systems. Systems which are connected in a point to point fashion residing at the

same level are called peer to peer systems. A network architecture involving both of the above types is called a hybrid architecture.

The various applications of networks are to share resources between devices like printers or USB's, CD's and various storage devices, transfer of data through emails or files, messaging and video conferencing.

Different types based on the geographical span are personal area network, local area network, metropolitan area network, wide area network. Among all the networks, the personal area network is the smallest one and it will be very personal to the user. The range of connectivity is around 10 meters. Examples are Bluetooth devices, wireless mouse and keyboard.

A local area network is spread across a building. It is usually operated by one single system, for example like a college or an office building. At least two and up to 16 million thousand devices can be present in a local area network. It enables resource sharing between the users. There may be local servers for storing files and does not have heavy traffic. It can be wired or wireless and uses Ethernet.

A metropolitan area network is usually spread throughout a city. For instance, cable television network. It can be implemented using Ethernet ring topology. This works between a local and wide network. Internet service providers provide metro Ethernet as a service. Wide area network will be a spread across a state or maybe an entire country. For instance, telecommunication networks are an example. These operate in a high data speed environment. Many companies can operate it together.

As these networks carry information which may have high value or tends to be secret, there is a need to protect it from possible intruders. Security mechanisms have been developed for this important requirement,

For the sensor network topology, we consider the unique model proposed by (Wagner, 2004). Figure 1 demonstrates our assumption for the considered model of the network in a WSN. The sensor nodes are partitioned into the disjoint pair of clusters, and each group has a group head which goes about as an aggregator. Information is intermittently gathered and collected by the aggregator. In this paper, we accept that the aggregator itself is not traded off and focus on algorithms which make accumulation secure when the individual sensor nodes may be compromised and may send false information to the aggregator. We expect that every information aggregator has enough computational power to run an IF algorithm calculation for information aggregation.

Literature Related to the Problem

To construct a secure technique in order to provide aggregation we need an algorithm with robust features. Such an algorithm ought to have two components.

1. Within the sight of stochastic errors, such algorithm should deliver estimates which are near the optimal ones in data theoretic sense. In this manner, for instance, if the noise shown or present in every sensor is a Gaussian autonomously disseminated noise with zero mean, then the estimate created by such an algorithm should have a difference near the Cramer-Rao bring down bound (CRLB) (Wasserman, 2012) i.e., it should be near the variance of the Maximum Likelihood Estimator (MLE). Regardless,

such estimation should be accomplished without providing to the algorithm the differences of the sensors, which are not available in practice.

2. The algorithm should similarly be robust within the sight of errors which are non-stochastic, for example, faults and malevolent attacks.

Trust and reputation frameworks have a noteworthy part in supporting the operation of an extensive variety of distributed frameworks, from remote sensor systems and online business foundation to informal organizations, by giving an appraisal of trustworthiness of members in such distributed frameworks (Roy et. al., 2012). A trustworthy evaluation at any given minute represents a total of the behavior of the members up to that minute and must be strong within the sight of different sorts of faults and malicious conduct. There are various motivating forces for attackers to control the trust and notoriety scores of members in a distributed framework, and such control can seriously impede the execution of a framework (Josang and Golbeck, 2009). The fundamental focus of malicious attackers is aggregate calculations of trust and notoriety frameworks (Wagner, 2004).

Algorithm 1: Iterative filtering algorithm.

Input: X, n, m.
Output: The reputation vector $\mathbf{r}$
$l \leftarrow 0$;
$\mathbf{w}^{(0)} \leftarrow \mathbf{1}$;
repeat
  Compute $\mathbf{r}^{(l+1)}$;
  Compute $\mathbf{d}$;
  Compute $\mathbf{w}^{(l+1)}$;
  $l \leftarrow l + 1$;
until *reputation has converged*;

Figure 2.1. Algorithm (de Kerchove and Van Dooren, 2010)

The Iterative filtering algorithm which is also known as an IF algorithm is proposed by de Kerchove and Van Dooren (2010). A collusion attack is introduced in wireless sensor networks to examine the vulnerability of the algorithm. Results of this algorithm can be applied to various other IF algorithms.

In order to understand this algorithm in a better way, consider a wireless sensor network which consists of n number of sensors(S). In figure 5, every aggregator reads one block of data at a time at m instants. Matrix X is used to represent the readings of the block. Here r is used to represent aggregate values for t number of instants. W is for weights of the sensor's trustworthiness. Rounds of iteration are performed to obtain the values of sensors consecutively.

Literature Related to the Methodology

De Kerchove and Van Dooren (2010) proposed in an IF calculation for processing the notoriety of items and raters in a rating framework. We quickly portray the calculation with regards to information collected in a WSN and clarify the helplessness of the calculation for a conceivable agreement assault. We take note of that our change is appropriate to other IF calculations as well.

In this paper, we utilize a Byzantine attack display, where the enemy can trade off a bunch of sensor nodes and infuse any false information through the bargained hubs (Awerbuch et. al., 2004). We accept that sensors are conveyed in a threatened and unattended condition. Therefore, a few hubs can be physically traded off. We accept that when a sensor hub is traded off, all the data which is inside the hub will be plainly open to the enemy or attacker. Consequently, we can't depend on cryptographic

techniques for escaping from the attacks, since the attacker may remove cryptographic keys from the traded off nodes. We accept that through the compromised sensor nodes the enemy can send false information to the aggregator with a reason for distorting the total values.

The attacker chooses the values in a very careful way to mislead the entire system of aggregation n-adversary model. This can be discussed in three scenarios. In the first scenario, the IF algorithm produces results very close to the actual expected value. Two sensor nodes are compromised in the second scenario and their readings are altered so that the average of these readings will be rounded to a lesser value. A collusion attack is launched by employing three compromised nodes in the third scenario. The attacked nodes report values different from the original values. It then instructs the third node to also report the false value, thus compromising the network.

Summary

The network model proposed by Wagner (2004) is discussed along with two important features an algorithm which is used for secure data aggregation should possess. The iterative algorithm on which collusion attacks is performed is also discussed briefly.

Chapter III: Methodology

The methodology undertaken for the study is discussed in this chapter. Details regarding the software environment in which the coding is performed are also discussed briefly.

Design of the Study

The approach followed in the study is qualitative in nature, as all of the algorithms were designed by researchers previously. The facts, as well as the algorithms were taken from various journals are collected and then implemented. A qualitative method is considered better than a quantitative method in this case because no surveys were conducted, and only existing facts and theories are studied.

Feasibility of the Study. It is important to know if the system that is planned is feasible to develop. This is the main outcome of the study being conducted. One should understand the problem before trying to solve it. To determine feasibility, usually a simple study is conducted with the help of people who are familiar with designing the process and analyzing the system.

There are three major areas one should consider determining the project's feasibility. They are: social, economical and technical feasibilities.

Social Feasibility. This part of the study is to check the level of system level acceptance by the client or user. It may also include the training sessions given to the user in order to make them comfortable with the system. It is the responsibility of the product developer to make the system acceptable to the client. Methods used to educate the user are important, as they help to ascertain the level of the user's

acceptance. Users should be made confident about the system and any criticism should be happily welcomed as they are the ultimate end users of the system.

Economic Feasibility. Economic feasibility endeavors to measure the costs of creating and executing the new system against the benefit that would accrue from having the new system set up. This study provides the upper management officials the need for new system implemented and its economic justification.

A simple and straightforward economic analysis which gives the real comparison of costs and benefits are considerably more significant for this situation. These analyses could also incorporate such factors as expanded consumer loyalty, changes in item quality, better ability of making decisions, assisting exercises, enhanced operations accuracy, better documentation and record keeping, speedier recovery of data, and better employee morale.

Technical Feasibility. This study is done to review the technical feasibility, which means the technical requirements of the system. Any system created must not place too great of a demand on the current accessible technical assets. This will lead to more demand on the accessible technical assets. This will create more demand levels being placed on the customers. The newly created system should have a simple requirement, as nearly zero changes are required for executing this system.

Data Collection

For the collection of data, various IEEE journals are referenced. Websites were also helpful related to computer networking and helped a lot to learn the basic definitions and to understand the complex algorithms.

The most widely recognized sources of information collection in qualitative research are meetings, observations, and survey of reports. The system is arranged and pilot-studied before the examination begins (Creswell, 2003) and places the information gathering strategies into four classes: perceptions, meetings, records, and varying media materials. It gives a compact table of the four strategies, the alternatives inside each option, the upsides of each sort, and the potential impediments of each.

We already noted that the researcher ordinarily has some sort of system (perhaps sub purpose) that decides and aides in developing ideas around the information gathering process. For instance, one period of the exploration may relate to the way in which experts and non-experts see different parts of a diversion. This stage could include having the competitor portray his or her view of what is occurring in a particular situation. Another period of the investigation may concentrate on the intelligent points of view and choices of the two gatherings of competitors while they are playing. The information for this stage could be gotten from recording them in real life and after that talking with them while they are viewing their performances on tape. Still another part of the investigation could be coordinated around the learning structure of the members, which could be dictated by a researcher-built instrument.

You should not anticipate that subjective information accumulation will be completed quickly. It is a time taking process. Gathering great amounts of information requires some serious energy (Locke et al., 2010), and fast meetings or short perceptions are probably not going to enable you to acquire a thorough understanding. On the off chance that you are doing subjective research, you should plan to be in the

environment for enough time to gather a great deal of information and comprehend the subtlety of what is happening.

Meetings. The meetings are without a doubt the most widely recognized source of information gathering in subjective investigations. The individual-to-individual arrangement is most predominant, yet once in a while aggregate meetings and center gatherings are also led. Meetings are typically run from an exceptionally organized style, in which questions are resolved before the meeting, instead of open-ended conversational configuration. In subjective research, the exceptionally organized arrangement is utilized fundamentally to accumulate sociodemographic data. Generally, be that as it may, interviews are more open ended and less organized (Merriam, 2001). Much of the time, the questioner solicits similar inquiries from every one of the members, yet the request of the inquiries, the correct wording, and the sort of follow-up inquiries may shift quite a bit impressively.

Being an effective questioner requires aptitude and experience. It was underscored before that the analyst should be able to create a rapport with the respondents. In the event that the members don't put stock in the scientist, they won't open up and give their actual emotions, considerations, and expectations. Full rapport will be set up after some time as individuals become more acquainted with and believe each other. A critical area of expertise in handling meetings is having the capacity to make inquiries in such a way that the respondent trusts that he or she can speak anything openly.

Kirk and Miller (1986) depicted their field of exploration in Peru, where they attempted to figure out how urban buttoned down white-collar class individuals thought about coca, the natural source of cocaine. Coca is legal and generally accessible in Peru. In their underlying endeavors to get the general population to educate them regarding the use of coca, they got the same socially affirmed answers from every one of the respondents. After they changed their style of questioning to making less touchy inquiries (e.g., "How could you discover you didn't care for coca?") did the Peruvians open up and expound on their insight into (and now and then their own utilization of) coca and even their own use of it. Kirk and Miller (1986) made a decent point about asking the correct inquiries and the benefit of utilizing different methodologies. Without a doubt, this is an essential contention for the legitimacy of subjective research.

Practice is necessary for skillful interviewing. Approaches to build up this ability incorporate recording your own particular execution in directing a meeting, watching experienced questioners, preforming simulations, and studying peers. It is essential that the questioner seem nonjudgmental. This can be troublesome in circumstances where the interviewee's perspectives are very unique in relation to those of the questioner. The questioner must be cognizant to both verbal and nonverbal messages and be adaptable in rethinking and seeking after specific questioning. The questioner must utilize words that are clear and significant to the respondent and must have the capacity to make inquiries with the goal that the member comprehends what is being inquired about. Most importantly, the questioner must be a decent listener.

The utilization of an advanced recorder is without a doubt the most well-known strategy for recording meeting information since it has the undeniable preferred standpoint of protecting the whole verbal record of the meeting for later examination. Albeit a few respondents might be apprehensive to talk while being recorded, this uneasiness ordinarily vanishes in a brief span. The fundamental downside with recording conversations is the potential failing of hardware. This issue is vexing and baffling when it occurs amid the meeting, yet it is especially problematic when it happens a short time later when you are endeavoring to replay and examine the meeting. Positively, you ought to have new batteries and ensure that the recorder is working properly ahead of schedule before the meeting. You ought to likewise stop and play back a portion of the meeting to see whether the individual is talking into the amplifier or microphone loudly and plainly enough and whether you are getting the information that you need. A few members (particularly kids) love to hear themselves talk, so playing back the chronicle for them can likewise fill in as inspiration. Keep in mind, in any case, that machines can breakdown whenever and have a backup plan.

Video recording is by all accounts the best strategy since you save what the individual said as well as his or her nonverbal conduct. The disadvantage to utilizing video is that it can be cumbersome and meddlesome. Along these lines, it is utilized rarely. Taking notes amid the meeting is another regular technique. Infrequently note taking is utilized as a part of the option to recording, essentially when the questioner wishes to take note of specific purposes of accentuation or make extra documentations. Taking notes without recording would keep the questioner from having the capacity to

record all that is said. It also keeps the questioner occupied with the note taking process, and not with her or his considerations and perceptions while the respondent is talking. In very organized meetings and when utilizing various sorts of formal instruments, the questioner would more be able to effortlessly take notes by confirming things and composing short reactions.

The minimum requirement for any favored system is endeavoring to recall and record subsequently information disclosed in the meeting. The disadvantages are numerous; however, this technique is utilized from time to time utilized.

Focus Groups. Another sort of subjective research strategy utilizes meetings on a particular theme with a small gathering of individuals, called a concentration gathering. This system can be productive on the grounds that the specialist can accumulate data around a few people in a single session. The gathering is normally homogeneous, for example, a gathering of understudies, an athletic group, or a gathering of educators.

In his 1996 book, *Focus Groups as Qualitative Research*, Morgan talked about the utilizations of center gatherings in sociology subjective research (Morgan, 1996). Patton (2002) contended that concentration of assembled members in meetings may provide quality controls since members have a tendency to act as a form of governance on each other that can serve to check false or outrageous perspectives. Center gathering interviews are normally pleasant for the members, and they might be less frightened of being assessed by the questioner in view of the gathering. The individuals

that are gathered get the chance to hear what others in the gathering need to state, and which may lead the people in the group to reconsider their own particular perspectives.

In the concentration aggregate meeting, the analyst isn't attempting to induce the gathering to achieve accord, rather It is a meeting to gain their insights. Taking notes can be troublesome, yet a sound or video recorder may take care of that issue. The quantity of inquiries that can be asked in one session is often restricted. Clearly, the concentration gathering ought to be utilized as a part of a bigger mix with other information gathering systems.

Observation. Observation in qualitative or subjective research for the most part includes investing in a drawn-out measure of energy in the setting. Field notes are taken all through the observations and are centered around what is seen. Numerous analysts likewise record notes to help with figuring out what the observed occasions may mean and to give assistance to noting the exploration inquiries amid resulting information examination (Bogdan and Biklen, 2007; Pitney and Parker, 2009). Although a few analysts utilize cameras to record what is happening at the exploration site, that strategy is not the norm, and most specialists utilize field notes to record what has happened in the setting.

One noteworthy downside to observation techniques is prominence. An outsider with a pad and pencil or a camera is attempting to record individuals' common conduct. Here stranger is the keyword. The work of a qualitative specialist is to ensure that the members get habituated with having the analyst (and possibly a device for recording)

around. For instance, the analyst might need to visit the site for no less than two or three days before the underlying information is accumulated.

Data Analysis

Tools and Techniques.

Java Technology. Java is an arrangement of many programming software and specs created by Sun Microsystems, and later bought over by the Oracle Corporation, that gives a framework to create/develop application programming and deploying it in a cross-stage user computing domain.

Java is utilized as a part of the processing stages from embedded equipment's/devices, cell phones to big business servers and even supercomputers. While less utilized, Java applets keep running in secure, sandboxed situations to give many highlights of local applications and can be installed in HTML pages.

The core of Java is the idea of a "virtual machine" that executes Java byte code programs. This byte code is the same regardless of what equipment or working framework the program is running under. There is a JIT (Just in Time) compiler inside the Java Virtual Machine, or JVM. The JIT compiler deciphers the Java byte code into local processor guidelines at run-time and reserves the local code in memory amid execution.

The utilization of byte code as an intermediate language grants Java projects to keep running on any platform that has a virtual machine accessible. The utilization of a JIT compiler implies that Java applications, after a short deferral while loading and once they have "warmed up" by being all or for the most part JIT-compiled, tend to keep

running about as quick as local projects. Java Runtime Environment is referred to as JRE. Since JRE, variant 1.2, Sun's JVM execution has incorporated a without a moment to spare compiler rather than an interpreter.

In spite of the fact that Java programs are cross-platform or independent of platform, the code of the Java Virtual Machines (JVM) that executes these projects isn't. Each supported working platform will have its own JVM.

Java can be considered as both a programming language and as a platform independent language. The Java programming can be referred to as a high-level language that can be described by the greater part of the accompanying trendy expressions sometimes called as buzz words:

- Simple
- Neutral of Architecture
- Object Oriented
- Portable
- Distributed
- High Performance
- Interpreted
- Multithreaded
- Robust
- Dynamic
- Secure

With many of the programming languages, we either compile or translate a program so you can run it on your PC. The Java programming language is bizarre in that a program is both compiled and translated. With the compiler, first you make an interpretation of a program into an intermediatory language called Java byte codes. The interpreter then parses and runs every Java byte code direction on the PC. Each java byte code instruction is parsed and ran by the interpreter in the computer. Compilation happens just once, whereas the interpretation will happen each time the program is executed. The accompanying figure shows how this functions.

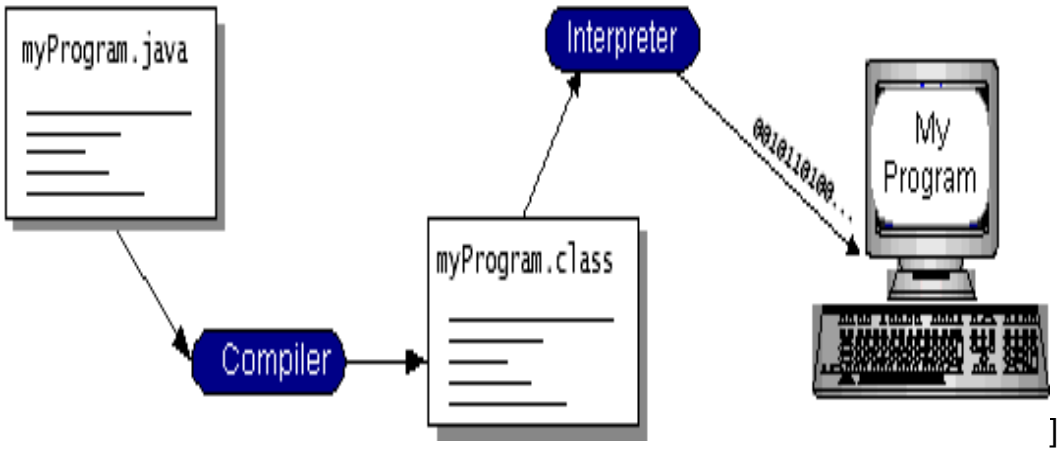

Figure 3.1. Java Program Compilation

NetBeans IDE is used for the development of the application. The entire application is developed in the java language and the corresponding data is stored in a MySQL database.

Java is the programming language and it keeps running on the stage also called as the platform, which is the JVM called as Java Virtual Machine. The Java programming language is a high-level language that can be described by the greater part of the accompanying trendy expressions. Java can be understood easily if the basics of Object Oriented Programming are known which makes it simpler to use. Java can be expanded as it is dependent on the Object model. OOP's or Object-Oriented Programming uses the concepts of classes and objects, where object is considered as instance of the class and exhibits a particular behavior. Class may contain many number of objects. The application that is developed on one operating system can run on any other operating system, then that application is considered to be platform independent. Java is platform independent but JVM is platform dependent.

Java bytecodes are the machine code instructions for the Java Virtual Machine. Every Java interpreter, whether it is a development tool or Web program that can run applets, is an execution of the Java Virtual Machine. Java bytecodes help make "compose once, run anyplace" feasible. The program can be compiled into bytecodes on any platform that has a Java compiler. The byte codes can then keep running on any JVM. That implies that as long as a computer has a Java Virtual Machine, the same program written in the Java programming language can work on any on Windows system, a Solaris workstation, or on a Mac.

The Java platform: A platform is the equipment or programming environment in which a system runs. The most prominent platforms like Windows 2000, Linux, Solaris, and MacOS are already known. Most platforms can be depicted as a mix of the working

system and the equipment. The Java platform contrasts from most other platforms in that it is not product dependent and that just keeps running on top of other equipment-based platforms. The Java platform has two segments which are the Java Virtual Machine and the Java Application Programming Interface (Java API). Java Virtual Machine has already been covered in-depth by now. It is the base for the Java platform and is ported onto different equipment based platforms. The Java API is an extensive gathering of instant programming parts that give numerous helpful abilities, for example, graphical user interface (GUI) plugins. The Java API is assembled into libraries of related classes and interfaces; these libraries are known as packages. The following segment highlights what usefulness a portion of the packages in the Java API provide. A native code will be code that after it is assembled, the ordered code keeps running on an equipment platform. As a platform-autonomous environment, the Java platform can be a bit slower than local code. Nonetheless, perceptive compilers which are very much attuned mediators, these bytecode compilers can very quickly convey execution near that of local code without undermining convertibility.

What can Java technology do? The most widely recognized sorts of projects written in the Java programming language are applets and applications. Surfing the net one might be familiar with applets. An applet is a project that sticks to specific traditions that permit it to keep running inside a Java-empowered program. Nevertheless, the Java programming language is not only used to write charming, engaging applets for the Web. The universally useful, high-level state Java programming language is

additionally a capable programming platform. Utilizing the library API, diverse types of projects can be created with it.

An application is a standalone program that works specifically on the Java platform. An exceptional sort of application known as a java server serves and backings customers on a system. Examples of servers are web servers, proxy servers, mail servers, and print servers. Another specific type of application is a servlet. A servlet can practically be considered as an applet that keeps running on the server side. Java Servlets are a favorite option for building intuitive web applications, while replacing the utilization of CGI (Computer Graphic Interfaces) scripts. Servlets are similar to applets in that they are runtime enlargements of applications. Rather than working in programs, however, servlets keep running inside Java Web servers, arranging or fitting the server.

Object Oriented Programming. Also referred to as OOP's is a programming language. Examples include java, C++. This model concentrates on objects instead of actions and logic instead of data. Example of objects includes an employee in an office who have different properties such as name, address, designation, qualification, salary.

Objects can be considered as user defined data types. They occupy space in the memory and will have an address in the memory. A program may contain many number of objects. The different objects present in the program sends message to each other for interaction.

The object contains data in the form of variables and methods in it which contains code to manipulate the data.

Classes do not take up any space in the memory. A class cam be declared using access modifiers: private, public, protected. By default, a class will be considered as private.

There are many concepts of object oriented programming such as: inheritance, polymorphism, abstraction, encapsulation.

Inheritance. It is the process through a class can acquire properties of another class. There are different types of inheritance such as single inheritance, multiple inheritance, multi-level inheritance, hierarchical inheritance.

Abstraction. Using data abstraction, in the program we expose only the necessary details to the world. The details which are trivial are hidden. For example, a car manufacturer will give details only about mileage but not about the underlying mechanism under it.

Data Encapsulation. In data encapsulation, data in the form of variables and methods or functions are wrapped in to a single unit. Here the variables are not accessible to the world, only the methods manipulating them can actually access their value. Hiding the data from being accessed is called as data hiding.

Polymorphism. It is the capability of the object to exhibit different forms. Polymorphism can be clearly understood through the concepts of overloading and overriding. It is widely used to implement inheritance in complex scenarios. Overloading is further divided into operator overloading and method overloading.

Message Passing. All the objects in the program communicate with each other continuously through messages. Typically, messages are requests from an object to other object requesting to execute a process.

Hardware and Software Environment.

Table 3.1. Hardware Requirements

Memory	4.0 GB
System	Pentium IV 2.4 GHz
RAM	512 MB
OS	Windows/Mac/Linux

Table 3.2. Software Requirements

Operating System	Windows XP/7
Coding Language	JAVA/J2EE
IDE	NetBeans 7.4

Computer assisted data analysis is performed. The results can be seen real time by running the computer program written in java programming language. The two scenarios with and without attack are analyzed in the discussion section in next chapter.

Summary

In this chapter, the study design approach used for the research, which is qualitative in nature is explained. Different types in which qualitative research can be performed is also elucidated. In the tools and technology section, the language used for programming Java is also discussed by giving a brief introduction of what it is and how code compilation occurs in such a high-level language. The requirements for the project, including both hardware and software are listed in the last section.

Chapter IV: Analysis of Results

This section deals with a clear simulation observation which studies the data aggregation method's robustness and efficiency. Determining the robustness and efficiency of the study in order to find the signal values basing on readings of the sensor is the objective of our analysis. This is examined in the case where faults or collusion attacks are present. The accuracy is calculated depending on the root mean square error which is referred to as RMS error. This estimation will be based on the iteration number required for IF algorithms convergence. Two scenarios are studied, one without any attack and the other one is with a collusion attack performed.

Data Presentation and Data Analysis

Many number of windows are developed using the java programming language. They are aggregators, senders, receivers and the base station. Screen shots of them are attached below.

There are two aggregators:

Aggregator 1- Data from receiver's one, two and three will be collected here.

Aggregator 2- Data from receiver's four and five is collected here.

Base station- It serves a hub for the wireless networks. It can either be a transmitter or a receiver.

Discussion

First the normal scenario without any collusion attack is performed. The following screenshots demonstrate that from figure 7 to figure 14. First, data will be sent from sender as shown in Figure 7.

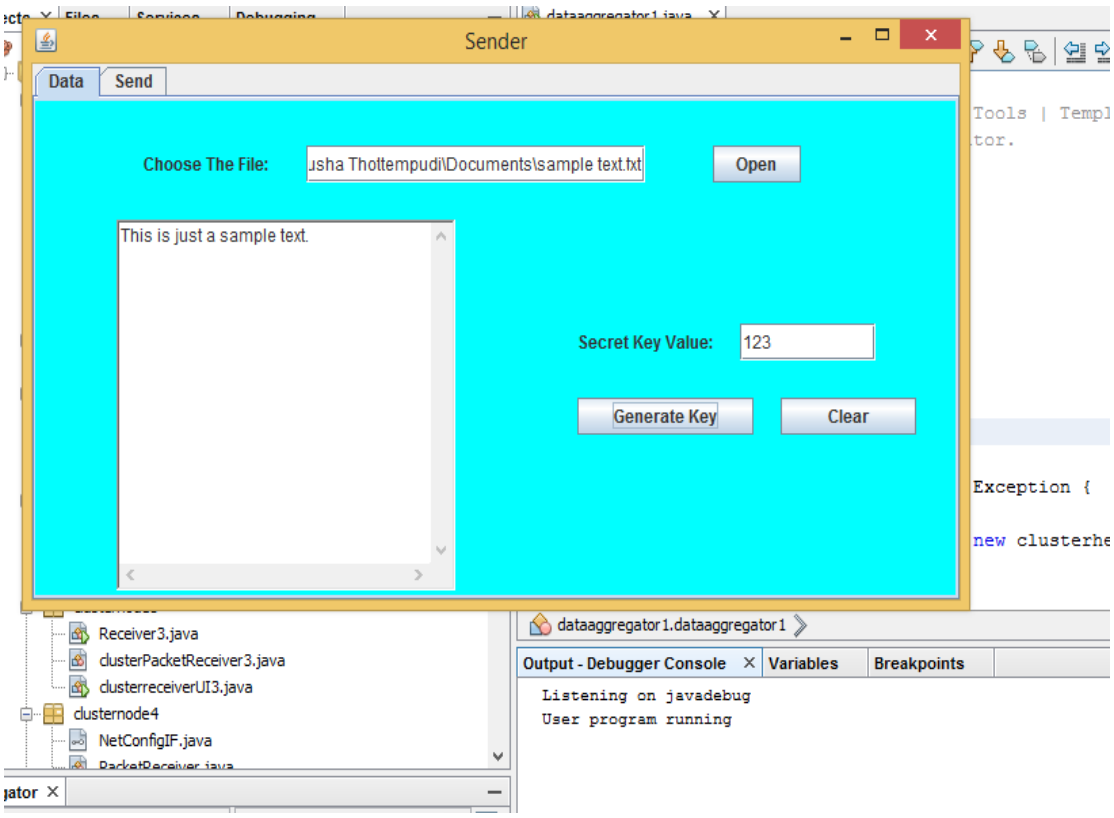

Figure 4.1. Sender with Text Uploaded

A text file with some sample text is selected. Click on the “Open” button. This prompts us to upload a file. Enter some value which can be considered as secret key value, and then click on the “Generate Key” button.

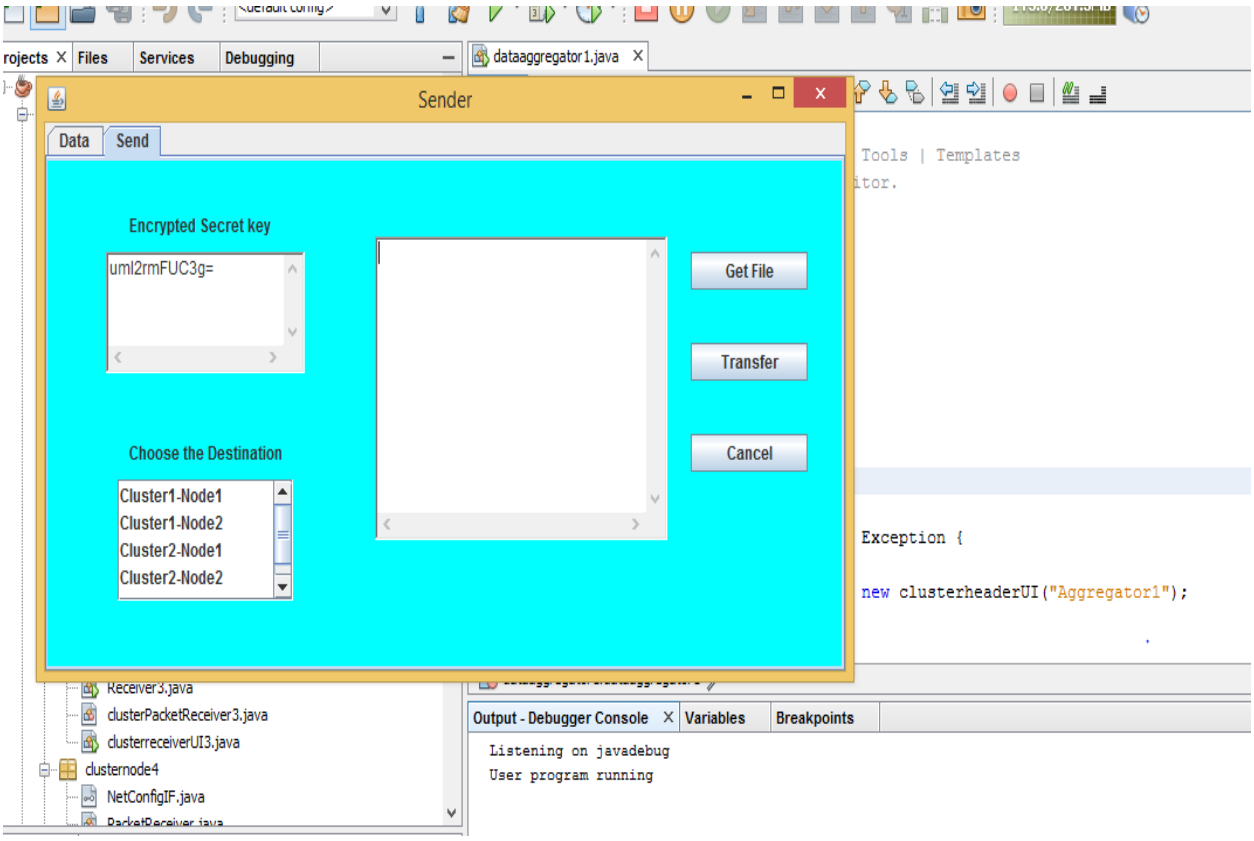

Figure 4.2. Sender with Encrypted Secret Key

Click on the send tab in the menu bar, now we can see an encrypted secret key has been generated. We can then select the destination by clicking on them. Here all the nodes are chosen. Click on the “Get file” button. This makes the sample text to appear in the text box. Then click on the “Transfer” button.

For instance, let us consider aggregator 1’s node and receiver 2. First, we need to enter the secret key in the data tab. Then click on the “Generate Key” button. Now in the receive tab, we can see the source side encrypted key and destination encrypted key. Click on the “Integrity” button. Then a message showing that a file has been received will be displayed.

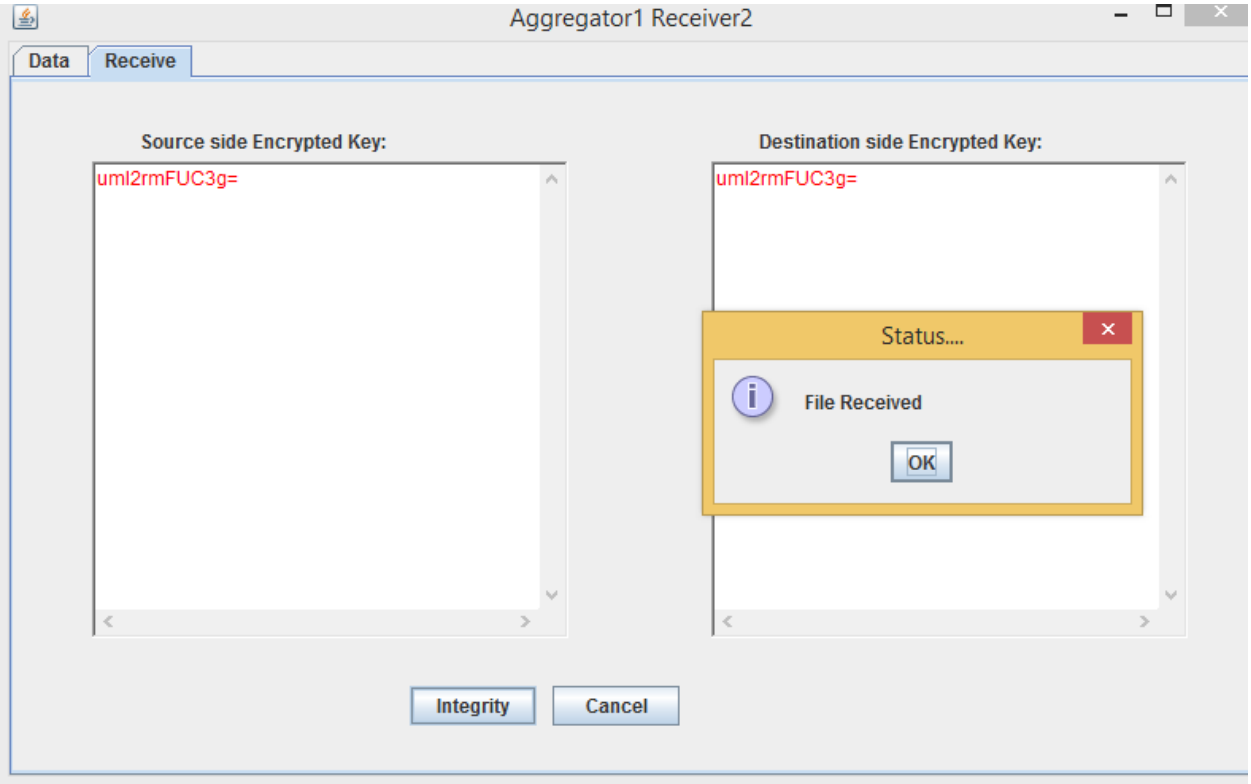

Figure 4.3. File Received Status in Receiver 2 - Aggregator 1

Click on the “OK” button to acknowledge receipt of the file. Now if the data tab is clicked, we can see the sample text in the received file text box.

The text will be received by receiver 1 and aggregator 1. The same way will be repeated for all of the nodes of aggregators- receiver 1 to receiver 5. This sample text data will be received by both of the aggregators. Aggregator 1 will receive the file and then this data will be aggregated from receiver 1, receiver 2 and receiver 3.

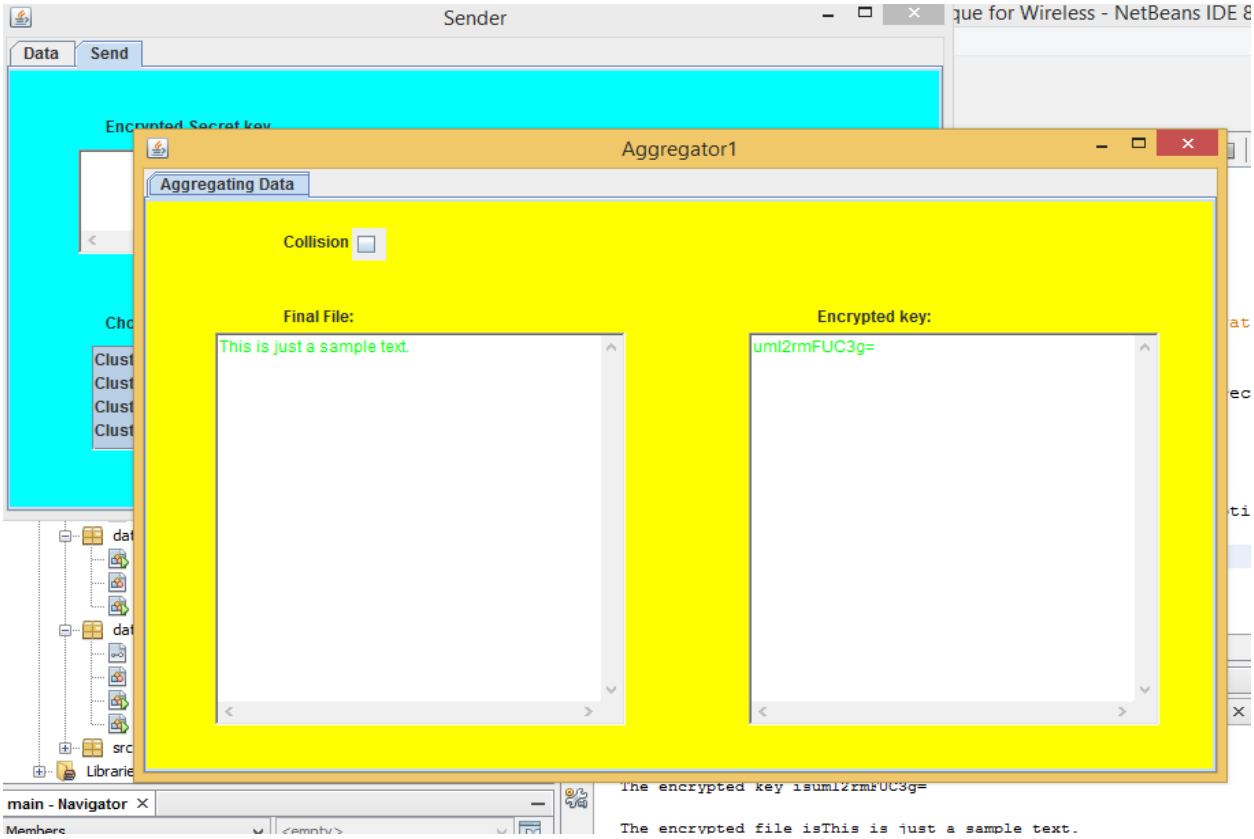

Figure 4.4. Aggregator 1 with File

In a similar way, aggregator 2 will then receive data from receiver 3 and receiver 4 of aggregator two. In the scenario above no attack was done and it showed a simple way in which data is transmitted in a wireless sensor networks.

Aggregator 1 -No collision attack is there in aggregator 1. So, collusion check box is not ticked

Aggregator 2- A Collusion attack is introduced in this aggregator.

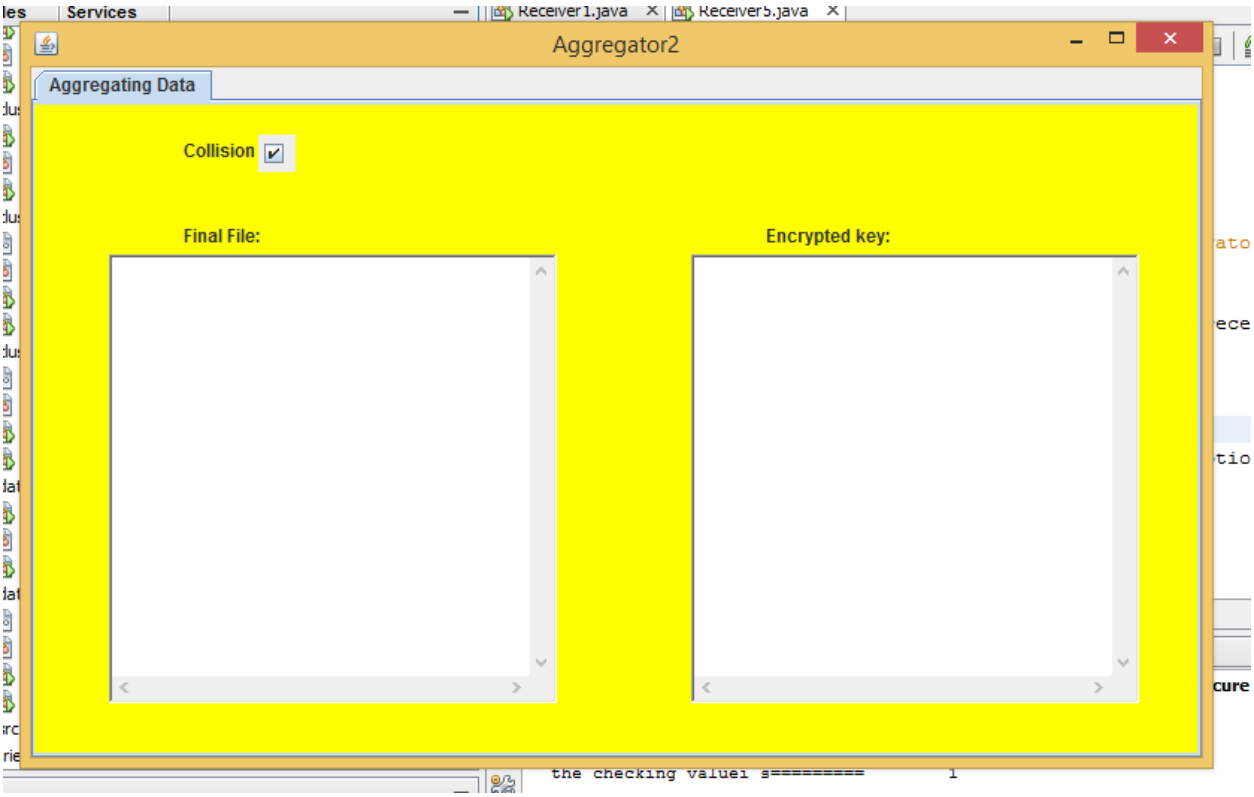

Figure 4.5. Aggregator 2 with Attack

Click on the “Collusion” check box in the window.

Sender. Receiver 1- Cluster node 1 is attacked. So, no data will be received. Enter a different key. No data will be received as the key value gets modified.

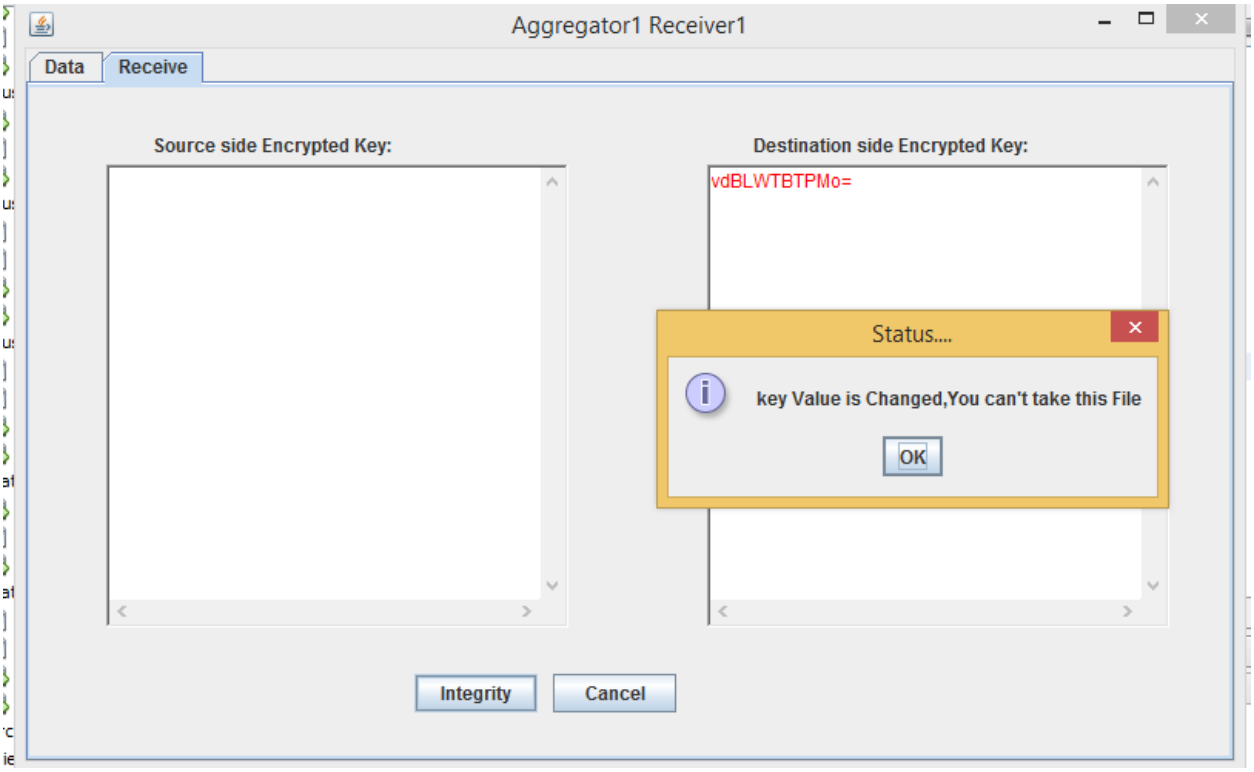

Figure 4.6. Key Value Changed

Receiver 2. Data will be received in receiver 2. We do not change key value for other receivers, so the file will be received in the usual way. Data will be received in the similar way for receiver 3 and 4. Receiver 5 will get a time's up message, as the key value has been changed.

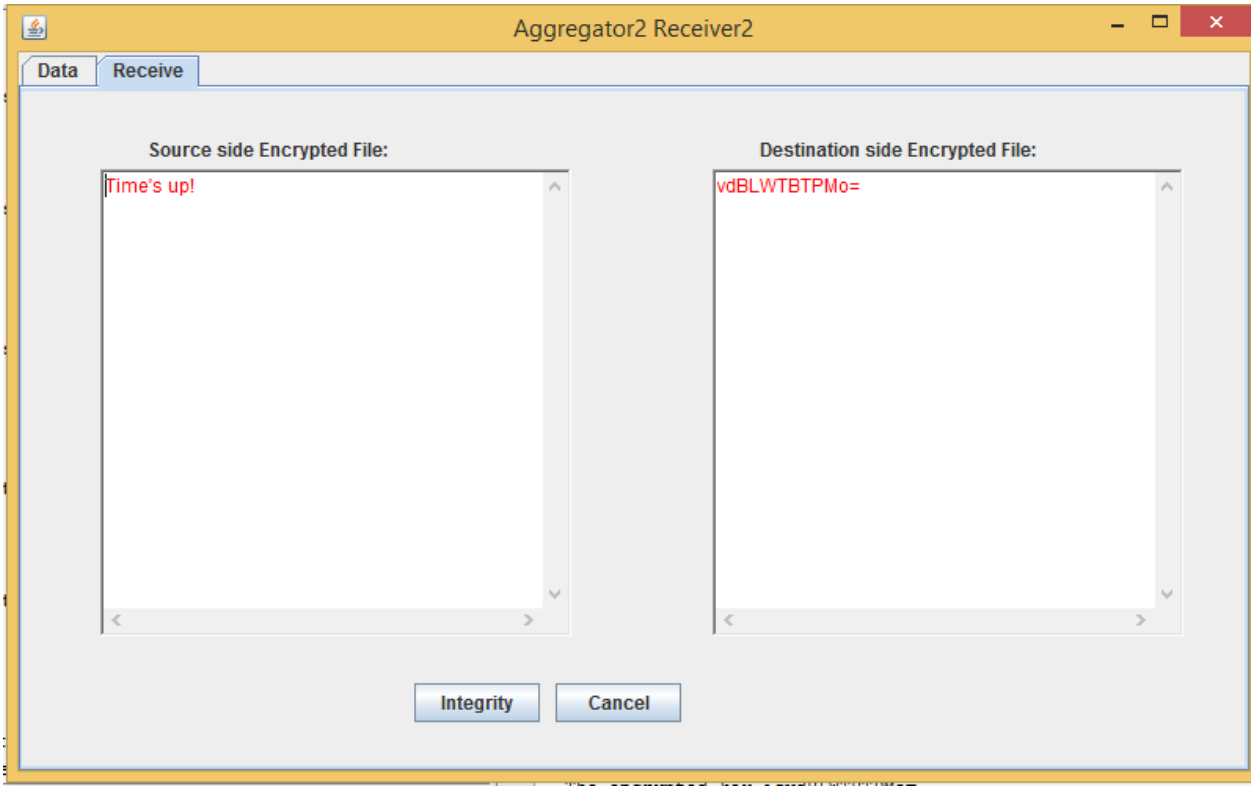

Figure 4.7. Receiver 5 Time's Up! - No File

Here, receiver 4 is Receiver1 in aggregator 2 and receiver 5 is Receiver2 in aggregator2. Aggregator 1 data will be received here, and the file appears in the aggregator 1 window.

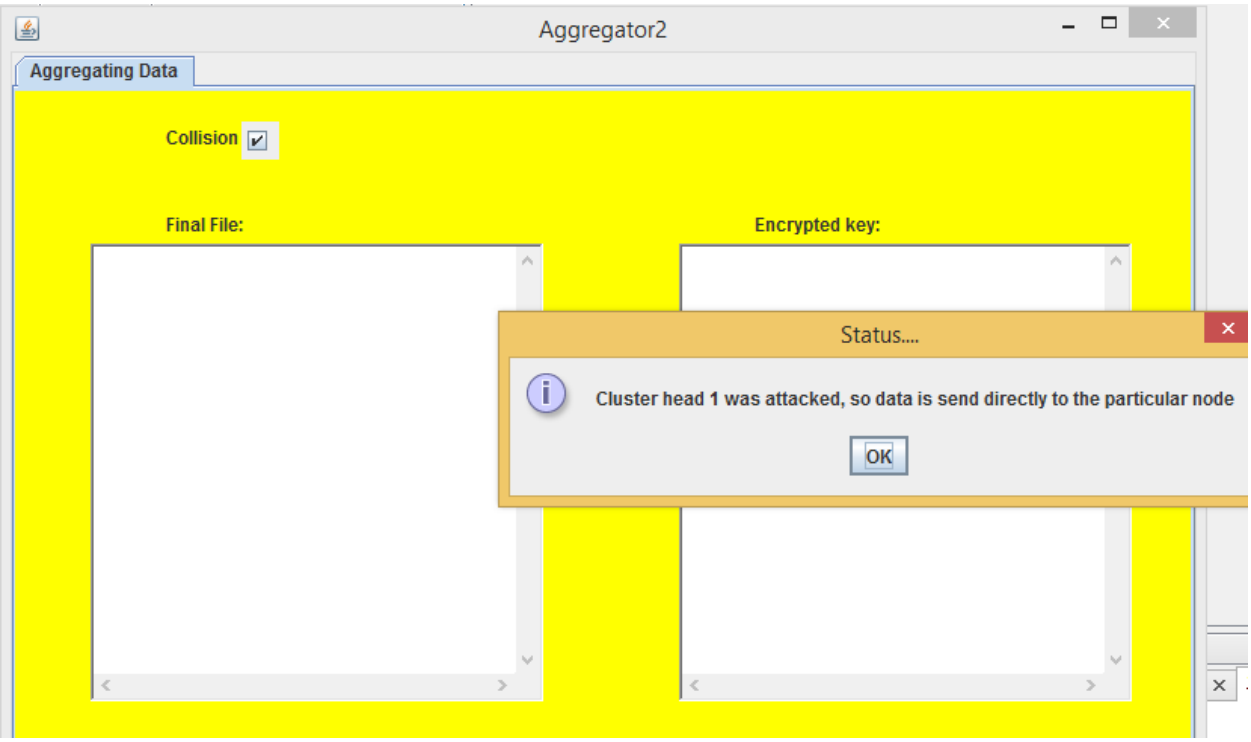

Figure 4.8. Collusion Attack on Node1

As the cluster node 1 is attacked, the data will not reach cluster node 5 which is receiver 2 of aggregator 2. So, no data comes to aggregator 2 because of receiver 2's corruption.

Results

The consequences of the study show that our attack situation is caused by the vulnerability found in IF algorithms which pointedly reduces the contributions of harmless sensor nodes which reports values that are close to average.

Summary

This chapter shows the study conducted in the form of screen shots and provides a brief analysis of results obtained. The figures depict both the scenarios: with a collision attack and without any attack. Data will be received by both aggregators only in

scenario one where there is no collusion attack. In scenario two, the collision attack is introduced with the cluster node one is attacked which tries to send malicious data. But due to the difference in key values, the time the aggregator notices the changes and will not take data from that particular node.

Chapter V: Conclusion and Future Work

In this chapter, a brief discussion about the study is provided. Plans for the future study are stated in the future work section. Applications such as submerged acoustic sensor frameworks, cyber systems based on sensors, time sensitive applications, and management of security can be built based on the future advancements of wireless sensor networks. There are some challenging areas where research should be focused, including power consumption which has a been continuous challenge for wireless sensor networks, we need to design algorithms which are capable of using the energy efficiently. Also, the cost of the hardware is another area which can be improved, small sensor nodes with low cost can and should be produced. However, the most important challenge is security where malicious data is inserted by compromising the nodes in the sensor networks. There may be numerous layers in a wireless network, so it is important to have standardization. Even though there are many organizations working on WSN's which follow different standards, most of the projects being worked on are in their initial stages. So, to resolve problems of non-conformity in the future, it is important to implement common standards so that all of the organizations can communicate and coordinate in a better way.

Discussion

There are various adversary models which are being studied to identify false data injection. A number of experiments are being conducted on robust data aggregation which is considered as a serious concern for wireless sensor networks. IF algorithms and data aggregation with security for detecting compromised nodes in the case of

wireless sensor networks, including trust and reputation systems in wireless sensor networks are the main three concepts dealt with in the research. Simple attacks like cheating by intruders are considered by present IF algorithms, but sophisticated attacks that involve more malicious cases like collusion attacks are not taken into consideration.

Conclusion

A minor collusion attack was introduced on the IF algorithms in this paper. All of these algorithms are existing ones. The algorithms were made more collision resistant by giving sensor node's initial trustworthiness approximation. This made the algorithms to be more accurate and gave the ability to converge faster.

Contributions of the Study

This paper provides a clear implementation of the novel collusion attack introduced on a network. The network consisting of nodes is developed in java platform, two scenarios without any attack and with attack are performed and the results are compared. Clear depiction of the steps is provided inn appendix.

Future Work

The future work will be related to protecting compromised aggregators in the way nodes are protected in this study. Another idea is to implement the same functionality in a sensor network which has already been deployed. These wireless sensor networks can be deployed in real time applications such as rural or forest areas. In forest areas, sensor networks can be used to detect animals to keep track of their movements. They can be used during fire incidents to safely evacuate people in a timely manner. Energy efficiency in the buildings can also be increased with the help of wireless sensor

networks. Deployments of wireless sensor nodes can be done in smart homes or offices and used by the in military to sense intruders and their activities as well in urban warfare environments.

References

[1]. S. Ozdemir and Y. Xiao, “Secure data aggregation in wireless sensor networks: A comprehensive overview,” Comput. Netw., vol. 53, no. 12, pp. 2022–2037, Aug. 2009.

[2]. C. de Kerchove and P. Van Dooren, “Iterative filtering in reputation systems,” SIAM J. Matrix Anal. Appl., vol. 31, no. 4, pp. 1812–1834, Mar. 2016.

[3]. E. Ayday, H. Lee, and F. Fekri, “An iterative algorithm for trust and reputation management,” Proc. IEEE Int. Conf. Symp. Inf. Theory,vol. 3, 2009, pp. 2051–2055.

[4]. C. T. Chou, A. Ignatovic, and W. Hu, “Efficient computation of robust average of compressive sensing data in wireless sensor networks in the presence of sensor faults,” IEEE Trans. Parallel Distrib.Syst., vol. 24, no. 8, pp. 1525–1534, Aug. 2013.

[5]. L. Wasserman, All of the Statistics: A Concise Course in Statistical Inference. New York, NY, USA: Springer, 2012.

[6]. D. Wagner, “Resilient aggregation in sensor networks,” in Proc. 2nd ACM Workshop Security Ad Hoc Sens. Netw., 2004, pp. 78–87.

[7] A. Jøsang and J. Golbeck, “Challenges for robust trust and reputation systems,” in Proc. 5th Int. Workshop Security Trust Manage.,Saint Malo, France, 2009, pp. 253–262.

[8] K. Hoffman, D. Zage, and C. Nita-Rotaru, “A survey of attack and defense techniques for reputation systems,” ACM Comput. Surveys, vol. 42, no. 1, pp. 1:1–1:31, Dec. 2009.

[9] B. Awerbuch, R. Curtmola, D. Holmer, C. Nita-rotary, and H.Rubens, “Mitigating byzantine attacks in ad hoc wireless networks,” Dept. Comput. Sci., Johns Hopkins Univ., Baltimore, MD, USA, Tech. Rep., 2004.

[10] C. de Kerchove and P. Van Dooren, “Iterative filtering in reputation systems,” SIAM J. Matrix Anal. Appl., vol. 31, no. 4, pp. 1812–1834, Mar. 2010.

[11] Shanika, C. 2015, www.winstudent.com/wireless-sensor-networks/.

[12] Creswell, 2003Research designs: Quantitative, qualitative and mixed method approaches.

[13] Merrium, S.B. 2001, Qualitative research and case study.

[14] Tarun Agarwal, 2014, www.elprocus.com/wireless sensor networks.

[14] S. Roy, M. Conti, S. Setia, and S. Jajodia, “Secure data aggregation in wireless sensor networks,” IEEE Trans. Inf. Forensics Security, vol. 7, no. 3, pp. 1040–1052, Jun. 2012.

[15] Locke, Silverman, Spirduso, “Reading and understanding research”, 2010

[16] Mohsen Rezvani, Aleksander Ignjatovic, Elisa Bertino, “Secure data aggregation technique for WSN in the presence of collusion attacks”, IEEE, Apr.2014.

[17] Kirk and Miller, Reliability and Validity in Qualitative Research, 1986.

[18] David L. Morgan, Focus groups as Qualitative Research, 1996.

[19] Patton, Qualitative Research and Evaluation Methods, 2002.

Appendix A: Sample Code

The sample code used for developing the application is provided for one instance for the data aggregator, sender, and receiver.

ClusterheaderUI.java

```
package dataaggregator1;
/*
   This class is used to provide UI for the Receiver node in the network
*/
import java.awt.*;
import java.awt.event.ItemEvent;
import javax.swing.*;
import javax.swing.table.*;
import java.util.Vector;
import java.awt.event.ItemListener;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
public class clusterheaderUI extends JFrame implements ItemListener{
private JTable           table       = null;
 private RouterTableModel tableModel   = null;
  public JTextField jt,jip,jtw;
  public JCheckBox check1;
```

```
 public TextArea jta,jta1,jta2;
 public JButton jb1,jb2,jb3,jb4,jb5,jb6,jbw;
 public JLabel jl,j2,j3,j4,jla,jlc1,jlc2,jlc3,jlc4,jlw,attack;
public String checkboxValue;
 JPanel jp=new JPanel();
 JPanel jp1=new JPanel();
JPanel jp2=new JPanel();
      JTabbedPane jtp=new JTabbedPane();
public clusterheaderUI(String title) throws Exception {
 jp1.setBackground(Color.YELLOW);
      jp.setLayout(null);
                  jp1.setLayout(null);
                  jt=new JTextField(20);
                  jip=new JTextField(20);
                  jtw=new JTextField(20);
                  jta=new TextArea();
             jta1=new TextArea();
             jta2=new TextArea();
             attack = new JLabel("Collision");
                check1 =new JCheckBox();
            jl=new JLabel("Secret Key Value:");
                 //j4=new JLabel("Choose The File:  ");
```

```
      jlw=new JLabel("Choose the destination:");
           //jb1=new JButton("Open");
           jb2=new JButton("Generate Key");
         jb4=new JButton("Integrity");
           jb5=new JButton("Cancel");
     jbw=new JButton("Request");
    jb6=new JButton("Clear");
j3=new JLabel("Final File:");
j4=new JLabel("Encrypted key:");
 jp1.add(jb4);
 jp1.add(j3);
 jp1.add(j4);
jp1.add(jta);
     // jp.add(jta1);
      jp1.add(jta2);
     jp1.add(jb4);
    jp1.add(jb5);
    jp1.add(check1);
      jp1.add(attack);
  // jtp.addTab("Data",jp);
           jtp.addTab("Aggregating Data",jp1);
    add(jtp);
```

```
            jta.setForeground(java.awt.Color.green);
            jta.setBackground(java.awt.Color.white);
            jta2.setForeground(java.awt.Color.green);
            jta2.setBackground(java.awt.Color.white);
            jta.setEditable(false);
            jta2.setEditable(false);
            check1.addItemListener(this);
          attack.setBounds(100,20,50,20);
             check1.setBounds(150,20,25,25);
              j3.setBounds(100,75,190,25);
             j4.setBounds(490,75,190,25);
             jta.setBounds(50,100,300,300);
             jta2.setBounds(440,100,300,300);
           setTitle(title);
           setSize(800,500);
         setLocation(100,100);
         setVisible(true);
  }
public void itemStateChanged(ItemEvent e) {
   int source=e.getStateChange();
  // System.out.println("the statechange value is"+ e.getStateChange());
    if (source == 1) {
```

```
            checkboxValue="1";
            System.out.println("the checking valuei s=========        1");
        } else
        {
          checkboxValue="2";
        System.out.println("the checking valuei s=========        2");
        }
 }
    public void addData(Vector data) {
      tableModel.addRow(data);
    }
    /*
       Inner and overriden class for table model
    */
    class RouterTableModel extends DefaultTableModel {
        RouterTableModel(){
         }
public static void main(String ar[]) throws Exception{
new clusterheaderUI("Aggregator1");
}
}
```

Dataggregator1.java

```
package dataaggregator1;
public class dataaggregator1 {
public dataaggregator1() throws Exception {
clusterheaderUI receiverUI = new clusterheaderUI("Aggregator1");
 while(true) {
        clusterheaderreceiver receiver = new clusterheaderreceiver(receiverUI);
      }
   }
   public static void main(String[] args) throws Exception {
      dataaggregator1 obj = new dataaggregator1();
   }
}
```

Receiver1.java

```
package clusternode1;
public class Receiver1 {
 public Receiver1() throws Exception {
 clusterreceiverUI1 receiverUI = new clusterreceiverUI1("Aggregator1 Receiver1");
      while(true) {
        clusterPacketReceiver1 receiver = new clusterPacketReceiver1(receiverUI);
      }
   }
   public static void main(String[] args) throws Exception {
```

```
      Receiver1 obj = new Receiver1();
   }
}
```

Sender.java

```
package Basestation;
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import javax.swing.border.*;
import java.util.Vector;
public class SenderUI extends JFrame implements ActionListener,ConfigIF
        {
    private  Container container      = null;
    private  JButton  closeButton    = null;
    private  JButton  sendButton     = null;
    private  JList  packetsList    = null;
    private  Vector   packetVector   = null;
    private  Sender   sender         = null;
    private  JLabel  routerIPLabel  = null;
    public BaseStationData send=null;
    public SenderUI() throws Exception {
```

```
    //    container = getContentPane()
        //setLayout to the container
    //    container.setLayout(new BorderLayout());
      //  sender = new Sender();
     //   Vector vector = getInput();
     getInput();
   /*      container.add("North",getLabelPanel());
        container.add("Center",getListPanel(vector));
        container.add("South",getButtonPanel());         */
        setSize(350,400);
        setLocation(150,150);
        setVisible(true);
    }
 public void getInput() throws Exception {
/*        Get the no of packets from the user
        String input = JOptionPane.showInputDialog( "Enter the Number Of Packets");
        int totalPackets = Integer.parseInt(input);
        packetVector = new Vector();
        for(int i = 0; i < totalPackets; i++){
           String packet = JOptionPane.showInputDialog( "Enter the Packet-"+(i+1));
packetVector.addElement(packet.trim());
        }
```

```
        return packetVector;                            */
 send=new BaseStationData();
   }
 /*   public JPanel getLabelPanel() throws Exception
                    {
         JPanel panel = new JPanel();
        panel.setLayout(new GridLayout(2,1));
        Font font = new Font("TimesRoman",Font.PLAIN,20);
        JLabel label = new JLabel("SENDER");
        label.setFont(font);
        label.setForeground(Color.blue);
        routerIPLabel = new JLabel("Source Router IP :"+routerIP);
        routerIPLabel.setForeground(Color.red);
        Border etched = BorderFactory.createEtchedBorder();
        font = new Font("TimesRoman",Font.PLAIN,13);
        Border border = BorderFactory.createTitledBorder(etched,
                "RSVP Sender",TitledBorder.LEFT,
                 TitledBorder.DEFAULT_JUSTIFICATION,font,Color.gray);
         panel.setBorder(border);
        panel.add(label);
        panel.add(routerIPLabel);
return panel;
```

```
    }
  public JPanel getListPanel(Vector vector) throws Exception
                    {
        JPanel panel = new JPanel();
        panel.setLayout(new GridLayout(1,1));
        packetsList = new JList(vector);
        JScrollPane scrollPane = new JScrollPane(packetsList);
        Border etched = BorderFactory.createEtchedBorder();
        Font font = new Font("TimesRoman",Font.PLAIN,13);
        Border border = BorderFactory.createTitledBorder(etched,
                "Generated Packets",TitledBorder.CENTER,
                 TitledBorder.DEFAULT_JUSTIFICATION,font,Color.red);
         panel.setBorder(border);
        panel.add(scrollPane);
       return panel;
}
public JPanel getButtonPanel() throws Exception
                    {
        JPanel panel = new JPanel();
        panel.setLayout(new FlowLayout(FlowLayout.CENTER));
       sendButton = new JButton("Send");
       closeButton = new JButton("Exit");
```

```
        Border etched = BorderFactory.createEtchedBorder();
        Font font = new Font("TimesRoman",Font.PLAIN,13);
        Border border = BorderFactory.createTitledBorder(etched    "",TitledBorder.LEFT,
TitledBorder.DEFAULT_JUSTIFICATION,font,Color.red);
        panel.setBorder(border);
        panel.add(sendButton);
        panel.add(closeButton);
       sendButton.addActionListener(this);
        closeButton.addActionListener(this);
    }  */
public void actionPerformed(ActionEvent event)
                    {
      try {
          if(event.getSource() == sendButton)
                              {
             routerIPLabel.setForeground(Color.green);
            // sender.sendPackets(packetVector);
             sender.closeConnection();
          }
          else if(event.getSource() == closeButton)
                              {
             System.exit(0)
```

```
}
      }
catch(Exception exception) { exception.printStackTrace();}
    }
  public static void main(String[] args) throws Exception {
SenderUI senderUI = new
SenderUI();senderUI.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
}
}
```

Appendix B: Screenshots of Individual Windows

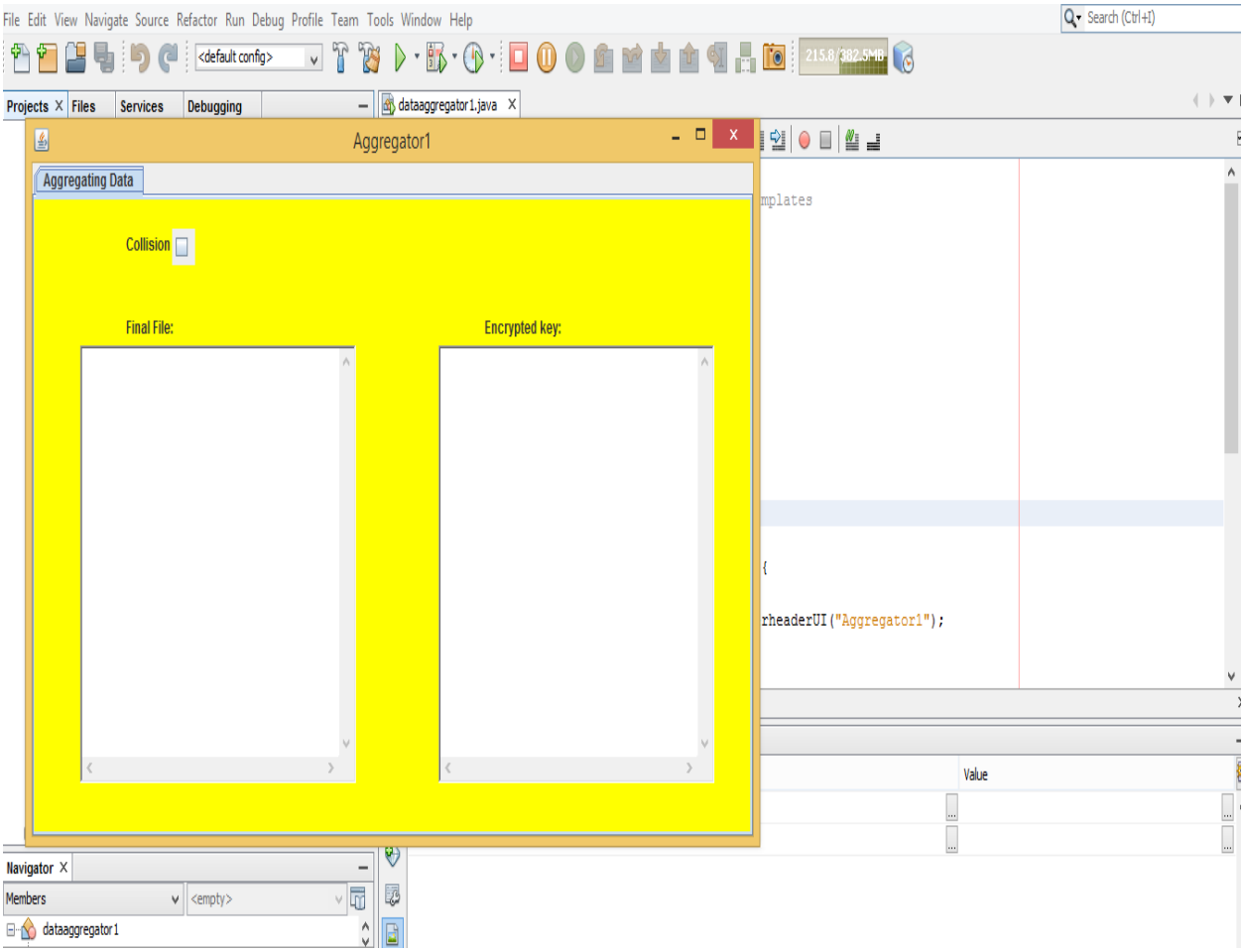

Figure B.1. Aggregator 1

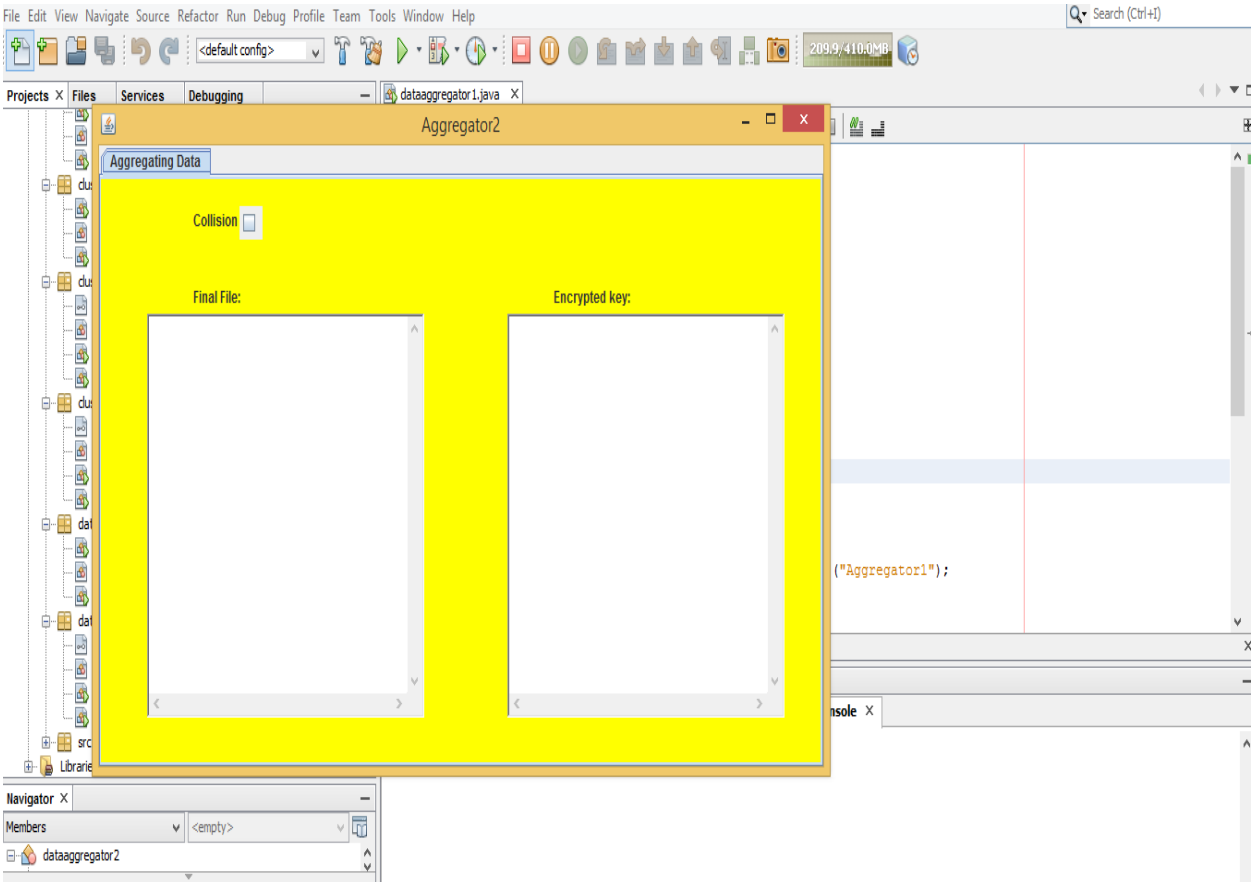

Figure B.2. Aggregator 2

The execution of the program needs all the windows to be opened beginning from sender to all the aggregators, receivers. Make sure none of the java application window closes before performing the attack.

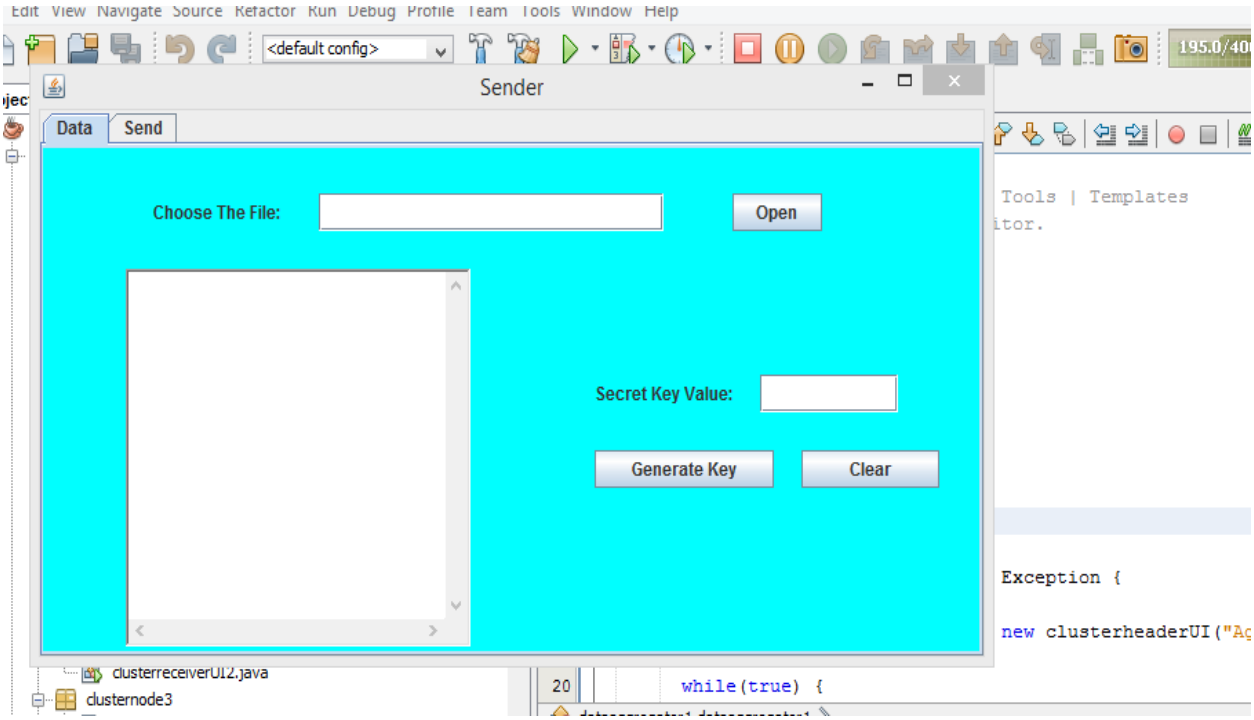

Figure B.3. Sender

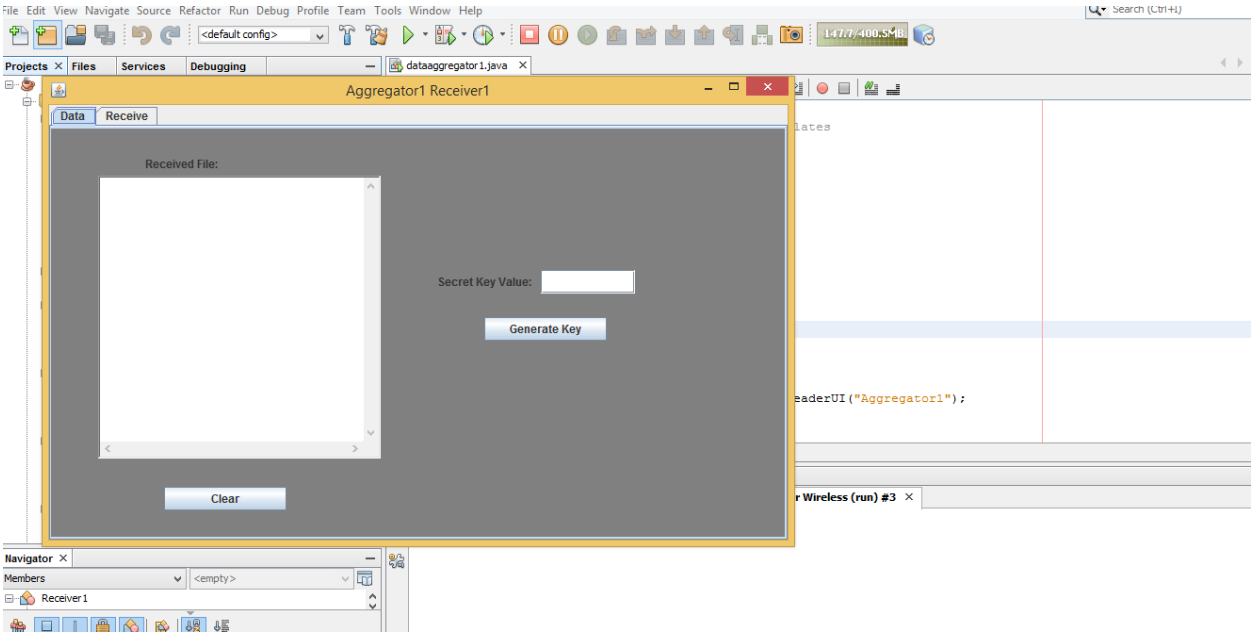

Figure B.4. Receiver 1 - Aggregator 1

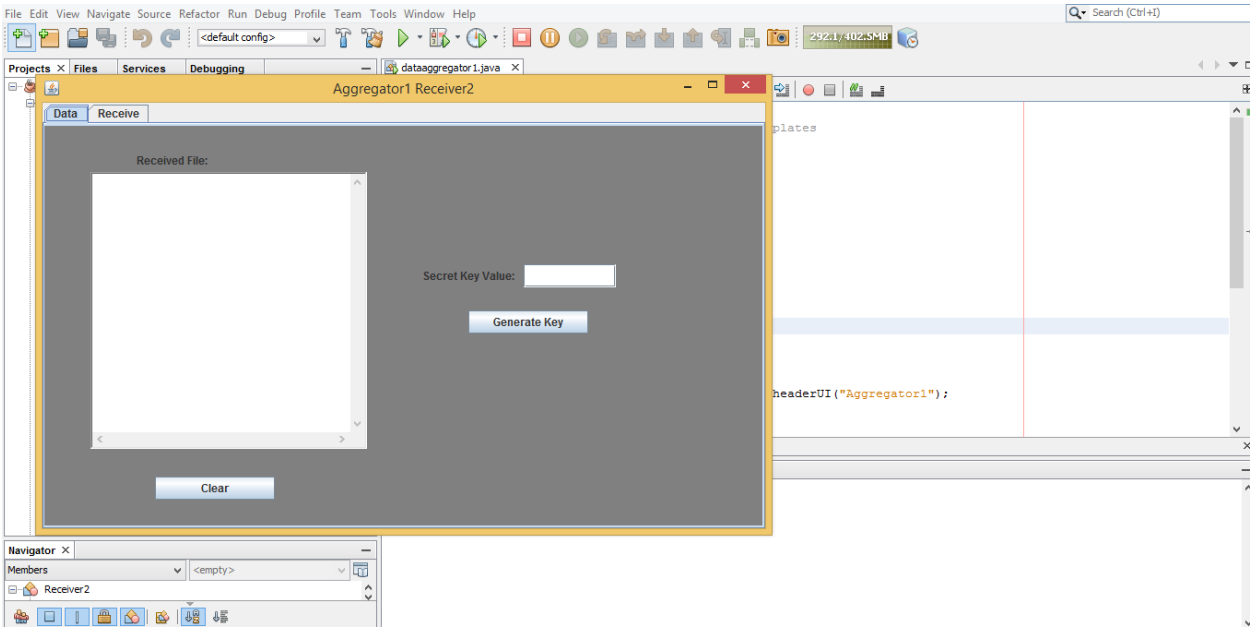

Figure B.5. Receiver 2 - Aggregator 1

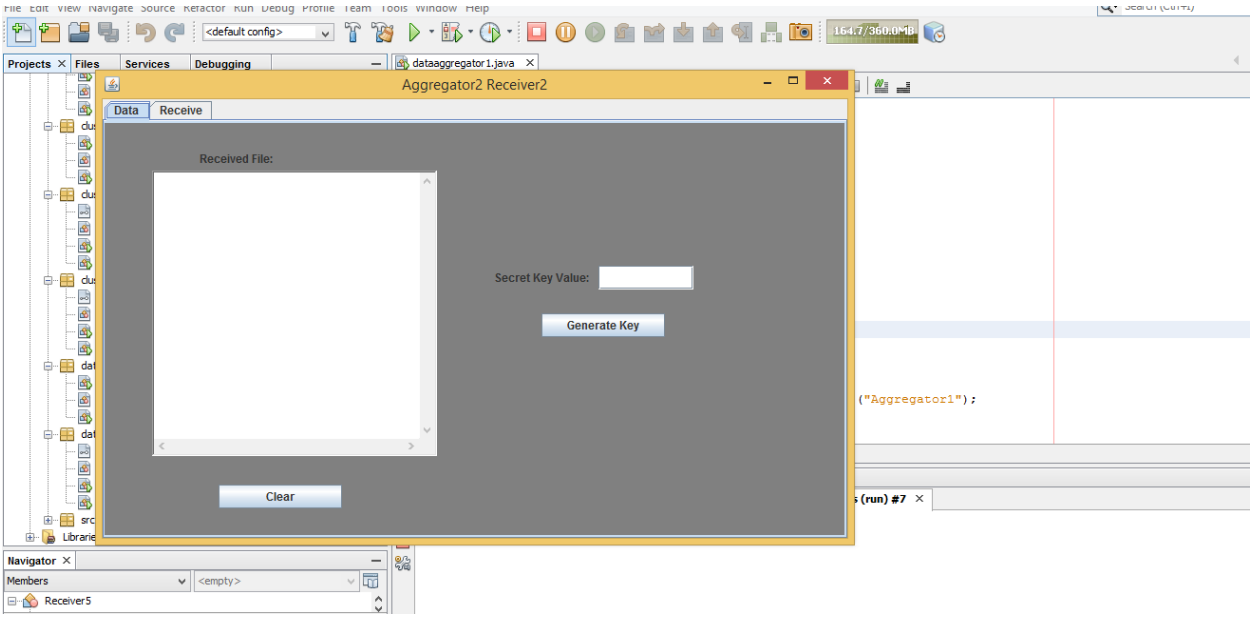

Figure B.6. Receiver 2 - Aggregator 2

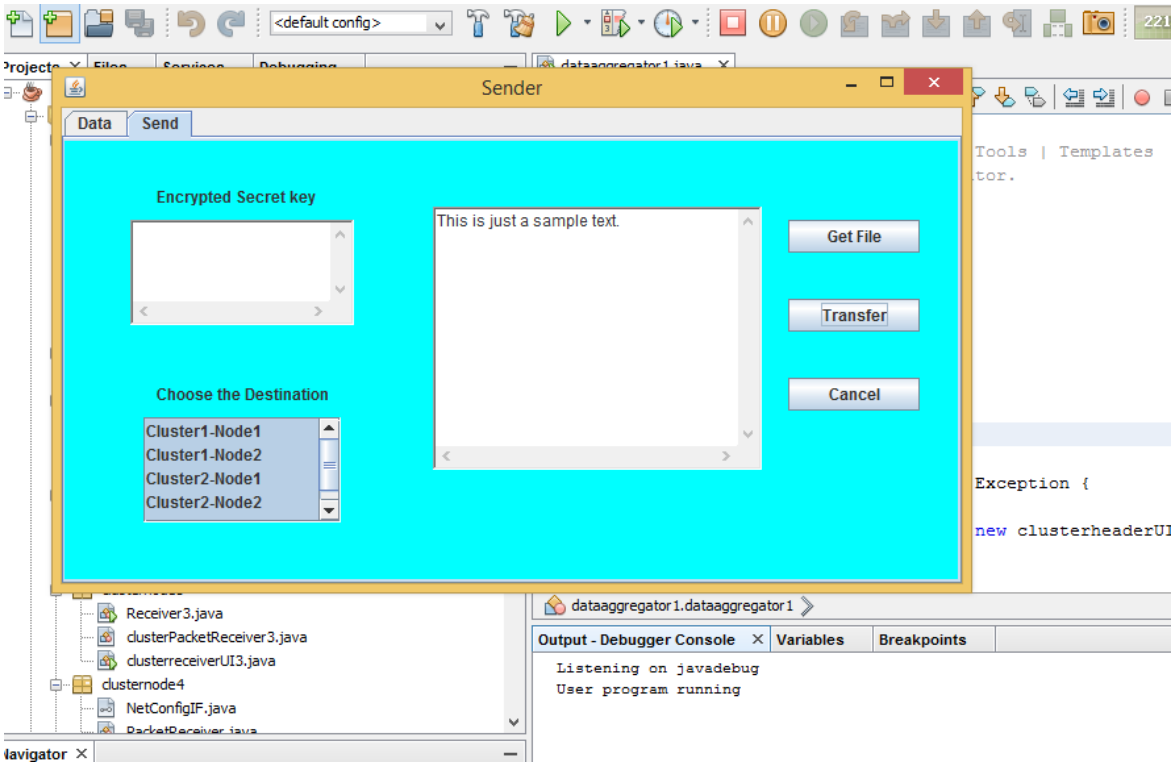

Figure B.7. Choose the Destination

Click on the “Get File” button and select a file from the desktop to upload. Make sure the file contains some data. Then click on the nodes to which the data should be sent. In the above figure 21, all of the nodes are selected. Then click on control key on the keyboard and select all the nodes. After node selection, click on the “Transfer” button. This will make the sender transfer data to all of the selected nodes. After this is done go to the respective nodes to check if the data has been received.

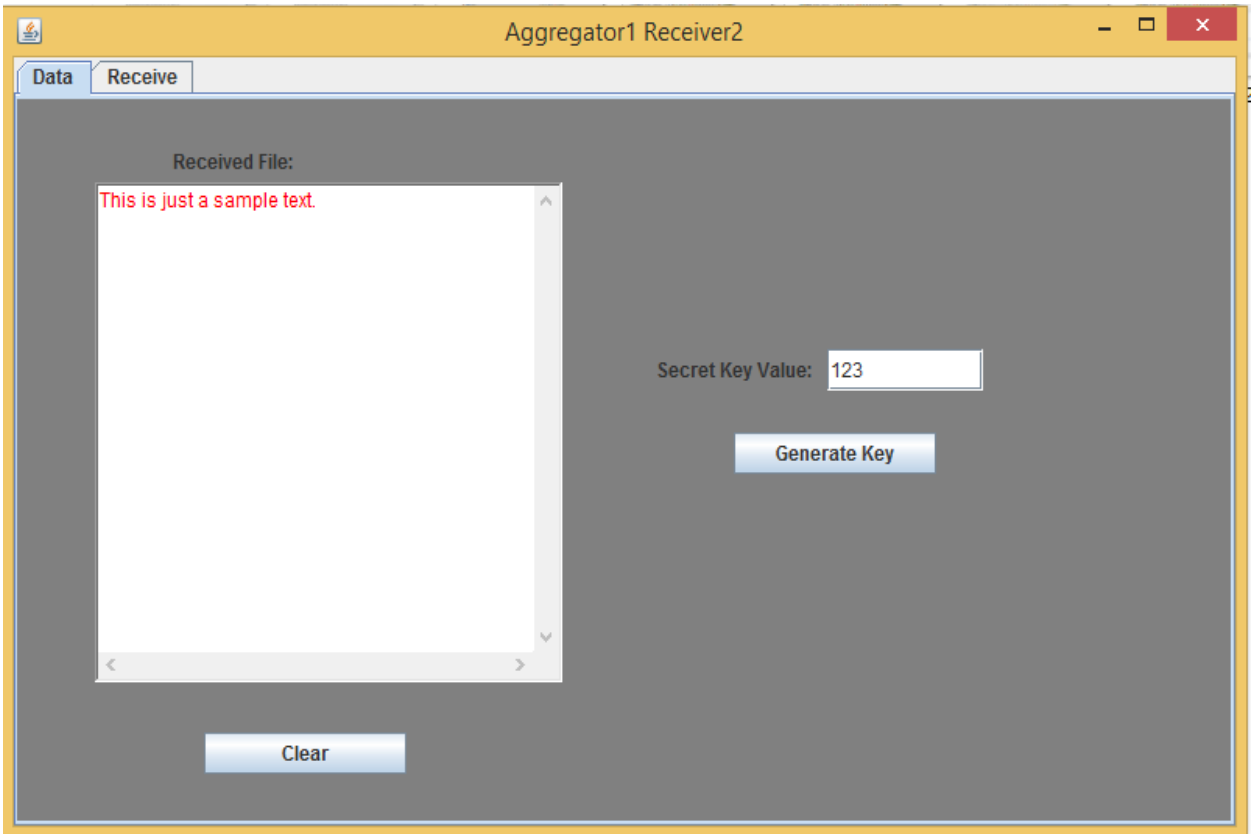

Figure B.8. Sample Text Received

Start with receiver 1, and enter the secret key value in the data tab. Click on the “Generate Key” button. If the keys match, the data will then be generated. Repeat the same procedure for all of the nodes.