CHAPTER 1

INTRODUCTION

1.1 GENERAL

In today’s highly technological world with rapidly increasing applications, wireless communication and networking play a consequential role. Usually, networking technology is an essential implementation for everyone’s day to day activities. Various individuals, different companies and organizations (which increasingly depend on electronic contrivances), used to process information and avail accommodations through a keenly intellective way in pervasive computing environment. It has proliferated to such an extent that seamless communication among contrivances, humans or between the two has become authentic, irrespective of mobility (Marius Senftleben et al 2008). In this context, the pervasive network environment has plenty of scope of consequential research and development off tardy (Anand Ranganathan & Roy Campbell 2008).

While designing a secure pervasive environment, issues such as privacy, trust and identity become more challenging for the designers. In this environment, in case of the security issues, the DDOS attack is the main problem in all ad-hoc scenarios (ie.) in MANET and in wireless sensor networks. Due to this dynamic nature, to combat or back track DDOS attack is difficult. In today’s internet, protecting victims from large scale bandwidth and target machines from heavy traffic and avoiding clogging of all the routes to the victim are difficult to implement (Marius Senftleben et al 2008). So, the
relevant information required by the victim is not possible to retrieve (Anand Ranganathan & Roy Campbell 2008). During the design of a privacy-sensitive pervasive monitoring system, issues like information misuse, leakage, information eavesdropping, social implications, designing privacy settings, lack of support in designing privacy-sensitive applications creep into the network layer and transport layer (Geetha Mariappan & Manjula Dhanabalachandran 2014). Also, the application layer may be affected by internal or external system attackers.

Privacy is also an important requirement of users in order to ensure that their personal or sensitive data is not used by others without their approval. Trust within or outside the system poses another challenge when the user crosses the boundaries of environment because in pervasive environment, trust relationship is defined by boundaries (Ameera Al-Karkhi et al 2012). Within the boundary, the normal authentication procedure facilitates the user. Practically, this will not support the user if they are outside the boundary. Therefore, privacy preservation techniques are used to define the relationship between privacy and trust.

1.2 BASIC PERVERSIVE COMPUTING ARCHITECTURE

Pervasive computing is the trend towards increasingly ubiquitous (another denomination is ubiquitous computing), connected computing contrivances in the environment (Ramesh Singh 2010). It is a trend that is being established by a convergence of advanced electronic – in particular, wireless - technologies and the Internet. Pervasive computing contrivances are not personal computers, but minutely diminutive contrivances. They can either be mobile or embedded in virtually any type of object such as cars, implements, appliances and sundry consumer goods; which communicate through interconnected networks (Ramesh Singh 2010). Researchers expect that in the future, keenly intellective contrivances all around us will maintain
current information about their locations, the contexts in which they are being utilized and germane data about the users.

The goal of researchers is to engender a system that is pervasively and unobtrusively embedded in the environment, thoroughly connected, intuitive, effortlessly portable, and perpetually available (Christian Makaya & Samule Lerre 2012). Among the emerging technologies expected to prevail in the pervasive computing environment of the future are wearable computers, perspicacious homes and keenly intellective buildings.

The goal of researchers is to engender a system that is pervasively and unobtrusively embedded in the environment, thoroughly connected, intuitive, effortlessly portable, and perpetually available. Among the emerging technologies expected to prevail in the pervasive computing environment of the future are wearable computers, perspicacious homes and keenly intellective buildings.

Pervasive computing is about enabling people to gain immediate access to information and accommodations anywhere, anytime, without having to scrounge for a phone jack (Ileladewa & Abdul Rahman 2013). However, while mobility and wireless technology are an astronomically
immense part of it, it is genuinely about making e-business personal. Thanks to the explosive magnification of the Internet, people will soon expect to be able to engage in electronic business effortlessly.

The PC is designed as a general-purpose platform that can run many different kinds of applications. Yet, it is found that individuals prefer separate appliances (Debashis Saha & Amitava Mukherjee 2003). Many people have a home PC as well as a separate fax machine and a stand-alone answering machine. Of course, in principle, one could utilize one's PC for faxing and answering the phone; it is probably even more frugal. Most people don’t utilize that because it is not convenient. A product that is tailored to a concrete task will always be more convenient than a general-purpose contrivance.

On the software side, Java could play a critical role as an interface standard. In case of infrastructure, there is an emerging area of "connectivity accommodations" that requires an incipient class of server. Different classes of contrivances will have different functions and capabilities (Brain Hall 2009). So, they will not be able to access the same kind of content. Rather than the indicted separate applications on the server side, the conception is to engender an intermediate server that will sit between the network computing content and the contrivances. It will take content indicted in HTML or XML, verbally express, and trans code it into a format for a particular contrivance (Brain Hall 2009).

Along with these incipient proxy or content connectivity, servers additionally provide other functions, such as sedulous assiduousness in the network. For example, if one drives into a tunnel and loses their connection as they are making an airline reservation, a proxy server will be able to consummate the transaction and then notify them later.
1.2.1 Sensor Networks

Sensor networks consist of collection of nodes capable of sense which control the environment, thereby allowing interaction between people, computers with the environment. The basic building blocks of sensor networks are radio tags which are context aware, sensible and monitor and report the events. These types of networks are generally wireless, self-powering and highly scalable. These networks are also self-organizing. Smart environments collect sensor data from the real world applications, which comes from multiple sensors and are distributed across locations.

Sensor networks are exploited to sense and monitor a remote area via sensor nodes and communicate this information to its peers or the main control unit. Sensors, controllers, devices and data are lodged into the physical world. As computer technology progresses in a rapid manner, it is possible to embed everything virtually with a sensor and store the data while integrating with other devices. Sensor networks are useful in scenarios where monitoring is either not feasible or not cost effective.

Challenges and opportunities of intelligent technologies are as follows:

- Using sensor networks improve decision making through enhanced information flow from distributed locations.
- Sensor networks improve visibility across the supply chain and real-time data availability for streamlined operations.
- Sensor networks connect and engage end customers accessing products and services via multitude of devices such as mobile, TV, sensors, appliances as well as via multitude of delivery channels such as wired/wireless internet, bluetooth etc.
1.2.2 Cloud Computing

Completing IT related services by reducing the cost urged the organizations to look for alternative models. This need for innovative solutions for reduced time paved way to a concept called Cloud computing. Organizations can compute and store data via cloud based platforms and services to leverage the cost. Cloud computing takes ubiquitous computing to the next level of technology. Cloud computing has the ability to offer computing power anywhere, at any time and on any device.

Cloud computing provides abstract infrastructure, dynamic allocation, scaling, movement of applications as well as commoditization of infrastructure. Cloud based applications are delivered over the internet and the pricing is based on consumption.

The features of cloud computing include lowered cost, faster time to market, high degree of flexibility, unlimited infrastructure growth capability, low lock-in, low cost of entry and low incremental cost. Cloud computing has been recognized quickly and the technology is being adopted rapidly. It is estimated that by 2012, 80 percent of Fortune 1000 enterprises will be paying for some cloud computing services, and 30 percent will be paying for cloud computing infrastructure services.

Some of the characteristic requirements for using cloud computing include applications that require storage and archiving, social community based applications as well as non-business critical applications. Cloud Computing based solutions, enable the enterprise to pay only for the server resources, applications and bandwidth that they use.

Some of the challenges as well as opportunities from leveraging cloud based technologies are:
Cloud computing optimization of costs by leveraging and consuming just in time computing and storage.

Launching of innovative solutions with reduced time to market for delivery, lower costs, reduced capex, pay-as-you-use and high scalability are the big features of cloud computing.

Cloud computing improves flexibility to scale up or scale down the business.

Cloud computing improves reliability and better business redundancy by leveraging distributed data centers.

1.2.3 Embedded Systems

An embedded system is the combination of computer hardware and software, either fixed in capability or programmable. Generally, embedded systems are designed for a particular kind of application device. Industrial machines, cameras, household appliances, automobiles, medical equipment, airplanes, vending machines and toys (as well as the more obvious cellular phones and PDAs) are among the myriad possible hosts of an embedded system. Programmable embedded systems are provided with programming interfaces, and embedded systems programming is a specialized occupation.

Certain operating systems or language platforms are tailored for the embedded market, such as Embedded Java and Windows XP Embedded. However, some low-end consumer products use very inexpensive microprocessors and limited storage, with the application and operating system, both parts of a single program. The program is written permanently into the system's memory in this case, rather than being loaded into RAM (random access memory) like programs on a personal computer.
The features of embedded systems are:

- Embedded systems are application specific & single functioned; application is known apriori, the programs are executed repeatedly.

- Efficiency is of paramount importance for embedded systems. They are optimized for energy, code size, execution time, weight & dimensions, and cost.

- Embedded systems are typically designed to meet real time constraints; a real time system reacts to stimuli from the controlled object/operator within the time interval dictated by the environment. For real time systems, right answers arriving too late (or even too early) are wrong.

- Embedded systems often interact (sense, manipulate & communicate) with external world through sensors and actuators and hence are typically reactive systems; a reactive system is in continual interaction with the environment and executes at a pace determined by that environment.

- They generally have minimal or no user interface.

Better monitoring of production processes to optimize operations, reduce costs, enhance production as well as prevent and/or detect health and safety issues.

1.2.4 Grid Computing

Grid computing is defined as a processor architecture that combines computer resources from various domains to reach some main objective. In
grid computing, the computers on the network can work on a task together, thus functioning as a supercomputer.

Typically, a grid works on various tasks within a network. It is capable of working on specialized applications. Grid is designed to solve problems that are too big for a supercomputer while maintaining the flexibility to process numerous smaller problems. Computing grids deliver a multiuser infrastructure that accommodates the discontinuous demands of large information processing.

A grid is connected by parallel nodes that form a computer cluster, which runs on an operating system, Linux or free software. The cluster can vary in size from a small work station to several networks. The technology is applied to a wide range of applications, such as mathematical, scientific or educational tasks through several computing resources. It is often used in structural analysis, Web services such as ATM banking, back-office infrastructures, and scientific or marketing research.

Carl Kesselman, Ian Foster and Steve Tuecke proposed the idea of grid computing in the early 1990s. They developed the Globus Toolkit standard, which included grids for data storage management, data processing and intensive computation management.

Grid computing is made up of applications used for computational computer problems that are connected in a parallel networking environment. It connects each PC and combines information to form one application that is computation-intensive.

Grids have a variety of resources based on diverse software and hardware structures, computer languages, and frameworks, either in a network
or by using open standards with specific guidelines to achieve a common goal.

Grid operations are generally classified into two categories:

- **Data Grid**: A system that handles large distributed data sets used for data management and controlled user sharing. It creates virtual environments that support dispersed and organized research. The Southern California Earthquake Center is an example of a data grid; it uses a middle software system that creates a digital library, a dispersed file system and continuing archive.

- **CPU Scavenging Grids**: A cycle-scavenging system that moves projects from one PC to another as needed. A familiar CPU scavenging grid is the search for extraterrestrial intelligence computation, which includes more than three million computers.

Grid computing is standardized by the Global Grid Forum and applied by the Globus Alliance using the Globus Toolkit, the de facto standard for grid middleware that includes various application components.

Grid architecture applies Global Grid Forum-defined protocol that includes the following:

- Grid security infrastructure
- Monitoring and discovery service
- Grid resource allocation and management protocol
- Global access to secondary storage and GridFTP
1.2.5 **Distributed Computing**

Distributed computing is defined as a computing concept that, in its most general sense, refers to multiple computer systems working on a single problem. In distributed computing, a single problem is divided into many parts. Each part is solved by different computers. As long as the computers are networked, they can communicate with each other to solve the problem. If done properly, the computers perform like a single entity.

The ultimate goal of distributed computing is to maximize performance by connecting users and IT resources in a cost-effective, transparent and reliable manner. It also ensures fault tolerance and enables resource accessibility in the event that one of the components fails.

The idea of distributing resources within a computer network is not new. This first started with the use of data entry terminals on mainframe computers, then moved into minicomputers and is now possible in personal computers and client-server architecture with more tiers.

A distributed computing architecture consists of a number of client machines with very lightweight software agents installed with one or more dedicated distributed computing management servers. The agents running on the client machines usually detect when the machine is idle and send a notification to the management server that the machine is not in use and available for a processing job. The agents then request an application package. When the client machine receives this application package from the management server to process, it runs the application software when it has free CPU cycles and sends the result back to the management server. When the user returns and requires the resources again, the management server returns the resources was using to perform different tasks in the user's absence.
In business enterprises, distributed computing has generally meant putting various steps in business processes at the most efficient places in a network of computers. In the typical transaction using the 3-tier model, user interface processing is done in the PC at the user's location, business processing is done in a remote computer, and database access and processing is done in another computer that provides centralized access for many business processes. Typically, this kind of distributed computing uses the client/server communications model.

The Distributed Computing Environment is a widely-used industry standard that supports this kind of distributed computing. On the Internet, third-party service providers now offer some generalized services that fit into this model.

More recently, distributed computing is used to refer to any large collaboration in which many individual personal computer owners allow some of their computer's processing time to be put at the service of a large problem. The best-known example is the SETI@home project in which individual computer owners can volunteer some of their multitasking processing cycles (while concurrently still using their computer) to the Search for Extraterrestrial Intelligence project. This computing-intensive problem uses your computer (and thousands of others) to download and search radio telescope data.

1.2.6 Grid Computing Vs. Distributed Computing

Since 1980, two advances in technology have made distributed computing a more practical idea, computer CPU power and communication bandwidth. The result of these technologies is not only feasible but easy to put together large number of computer systems for solving complex computational power or storage requirements. But, the numbers of real
distributable applications are still somewhat limited, and the challenges are still significant (standardization, interoperability etc).

As it is clear from the definition, traditional distributed computing can be characterized as a subset of grid computing. Some of the differences between these two are as follows:

- Distributed Computing normally refers to managing or pooling the hundreds or thousands of computer systems which individually are more limited in their memory and processing power. On the other hand, grid computing has some extra characteristics. It is concerned to efficient utilization of a pool of heterogeneous systems with optimal workload management utilizing an enterprise's entire computational resources (servers, networks, storage, and information) acting together to create one or more large pools of computing resources. There is no limitation of users, departments or originations in grid computing.

- Grid computing is focused on the ability to support computation across multiple administrative domains that sets it apart from traditional distributed computing. Grids offer a way of using the information technology resources optimally inside an organization involving virtualization of computing resources. Its concept of support for multiple administrative policies and security authentication and authorization mechanisms enables it to be distributed over a local, metropolitan, or wide-area network.
ASPECTS OF PERVERSIVE COMPUTING AND ISSUES

Security is withal going to be key and Standards will be critical for pervasive computing.

The current phase of pervasive computing, in which computers are already being embedded in many contrivances, can be thought of in sundry ways. There are four major aspects of pervasive computing that appeal to the general population:

- Computing is spread throughout the environment
- Users are mobile
- Information appliances are becoming increasingly available
- Communication is made more facile-between individuals, between individuals and things, and between things

Computers will not only be increasingly mobile, but information will be accessible from any mobile position. Pervasive computing is all about access to your information, anytime, anywhere, from any contrivance.

Today computing is already embedded in more places than just our desktop computers (Ark & Selker 1999). Computers make our cars run congruous with antilock braking systems and power steering. These examples illustrate what seamless computing should be-it can provide sublime functionality without requiring that the utilizer understand its inner workings (Ark & Selker 1999).

While designing secure pervasive environment the following designs are to be taken as consideration:
• System reliability
• System confidentiality
• Infrastructural network
• Personal devices
• Increasing intra connectivity between devices
• Risks involved in the environment
• Privacy of sensitive information
• User’s data protection from the attacker
• Trust relationship with in the system
• Finding trust boundaries between external entities
• Secure Policies
• Trusted third party
• Dynamic access rights

In pervasive environment, users can access any resources and services from anywhere at any time leading to different serious risks. A network inside the pervasive environment is dynamically changing their topology according to the nodes joining to it. Firewall is used to protect such an environment from untrusted and unauthorized users. This will verify and separate the users as trusted parties and non-trusted parties. This is not workable, if the device stolen by somebody. In the same way devices can exchange their personal information which are stored in it. When devices are in a single domain, privacy is not a big problem. It is crucial to develop the framework for different domains. The huge amount of information stored in
the devices or environment should not be tracked by unwanted third parties. So the privacy of information is related to trust and security. This is shown in the Figure 1.2.

![Figure 1.2 Challenges in Pervasive Computing](image)

Computers will not only be increasingly mobile, but information will be accessible from any mobile position (Anders Oission 2005). We should not have to carry around contrivances containing our information. Rather, contrivances will apperceive who we are and obtain information about us, through “remembrance agents” or adaptive utilizer models, Internet information storage, or other designates (Anders Oission 2005).

Information appliances have human-computer interfaces. An information appliance should be facile for anyone to utilize and the interaction with the contrivance should be intuitive. Careful design is critical for an intuitive interaction with the contrivance. Although the desktop computer can do many things, this functionality can be disunited into more congruous contrivances.
1.4 OBJECTIVE OF THE WORK

Nowadays, the security challenges faced in the pervasive computing environment are very high. In order to provide a secure reliable communication between user and system environment there is a need to develop an enhanced privacy model.

There are different mechanisms in pervasive environment, have been developed for providing security of data during communication and preserving privacy of data stored in memory by many researchers. Several researchers presented their views and ideas and improvements for issues related to trust, security and privacy. Though there may be more research papers in the area of pervasive environment with trust, security and privacy, still there is scope for development of new mechanisms for providing security, preserving privacy with dynamic trust management system.

The main objective of the security model focuses on preventing various attacks that occur in pervasive environment at different OSI layers. This model helps in increasing the reliability by avoiding duplicate packets not to be forwarded and each node maintains a cache. It also helps to develop cross layer security mechanism to the environment. The main focus of this model is to develop effective, efficient, creative and comprehensive prevention, detection and response mechanism for addressing the problems of DDOS flooding, during and after an actual attack. The behavior of this mechanism is going to compare with an existing protocol like AODV.

In privacy model, the main objective is to provide privacy mechanisms with the help of qualitative and quantitative models.

1. The quantitative approach can be applied in small size of infrastructural facilities available in the environments
considering the privacy handled by many covered entities, the amount of acceptable uncertainty in each and every exchange and by developing the techniques based on the behavior of the entities involved in the network. This approach focus on the techniques of data globalization, data virtualization and data embellishment. This method handles the data in which system want to preserve the privacy before sending it to the requester. The first technique (data globalization) means to replace the original contents by its less specific but semantically consistent. So, attackers cannot alter the original image of the data. The second method data virtualization is the technique to provide the amount of data for accessing the information. It also develops the transformation on the contents of the table in which those meta data going to handle by multiple consumers. So, this technique provides a virtualized table to the multiple consumers without any replication of data. The last method (data embellishment) means to decorate the original contents of information with another form by either diminishing or elaborating it. So, the attackers cannot get original sensitive information. At last, this approach provides privacy preservation by applying these specified techniques to the data store to develop Meta data specification.

2. The main goal of qualitative approach is to protect the sophisticated data being modified by unauthorized user by developing the techniques based on the system metrics of trust, privacy and security. Because unless otherwise specification of mathematical model, any one can’t get single solution for any real world problem. So there is a need for finding single solution for the problem of privacy preservation with metrics of this environment. This approach is to develop such a model for trust, privacy and security
in pervasive healthcare. At end of this work, these specified approaches are going to be compared with an existing systems by clinical trials.

As pervasive computing environments can come in different formats such as static networks (e.g. sensor network), mobile networks (e.g. MANET) and wireless networks (e.g. Pure and Managed), the requirements given by the security and privacy issues play a major role, which can be solved by mechanisms mentioned above for maximum level of satisfaction in the network.

1.5 THESIS ORGANISATION

The research work carried out is presented in this thesis over five chapters and its organization is presented as follows:

Chapter 1: The first chapter in this thesis highlights the importance of ubiquitous computing and brings out the challenges in the design of secured environment and their deployment. The motivation for taking up the research work is carried out in this thesis and the specific objectives of the research work are also highlighted.

Chapter 2: A detailed literature survey is presented in this chapter. Some of the relevant works associated with the issues like security, trust and privacy of the research work in this thesis are explained in detail and the mechanisms for solving those issues are described. The security mechanism that have been considered is Ad-hoc On-demand Distance Vector (AODV) as the base algorithm and Dynamic trust and security mechanism is proposed for providing enhanced privacy. A g-anonymity technique is also proposed based on non-knowledge base to preserve the same privacy in healthcare system.
Chapter 3: This Chapter presents the frame work for preserving privacy with dynamic trust management policies for enhancing security in a ubiquitous computing environment. It has 2 modules: (1) Security Module, (2) Privacy Module.

Chapter 4: The concept of Dynamic trust and security mechanism is proposed. This chapter describes all the possibilities of attacks that occur during the communication in pervasive environment. Based on protocol level, DDOS flooding attacks are categorized into Network or transport level attacks and application level attacks. This chapter also describes defense mechanism to deal with these attacks in particular, how the traffic will be reduced by sender at sender side. The performance evaluation of the routing protocol AODV with extended system is carried out here to improve the metric of reliability in pervasive environment with one way communication and multiway communication to achieve the privacy, trust and security.

Chapter 5: The privacy preservation techniques presented in this chapter are g-anonymity that is based on Data Globalization, Data Virtualization and Data Embellishment. The privacy mechanism developed in this study enables both micro data and macro data to dynamically manage information privacy in pervasive computing environment and to define health care policies for all entities considering their trust and corresponding attributes. This mechanism also offers protection against privacy threats existing in pervasive computing environments. The quantitative method is also proposed for trust, security and privacy in ubiquitous healthcare system. A model based metrics for U-health care system is identified. The various trust relationship between the covered entities of healthcare information is considered and its metrics based on the role and the information handled are arrived. The privacy of the information in ubiquitous healthcare system is
derived based on mode of information, deployment and its operational environment.

**Chapter 6:** The conclusions based on the research work are carried out and simulations of the proposed protocols using NS2 are presented and discussed in this chapter. The following observations are made based on the results obtained.

The number of reliable paths increases while the number of nodes increases in both single and multi way communication. Hence the nodes can be able to send or receive the packets in less time because of reduction of malicious attacks and also traffic.

The privacy mechanism developed in this g-Anonymity enables both micro data and macro data to dynamically manage information privacy in pervasive computing environment and to define health care policies for all entities considering their trust and corresponding attributes in this chapter.