2.1 INTRODUCTION

There are so many security threats available in pervasive computing such as jamming attacks, masquerade attacks and DoS attacks. To provide security, a detailed literature review has been carried out to provide a security framework with the best cryptographic method that best suits a cloud based pervasive computing environment. A detailed literature review has been carried out in this chapter relating to various cryptographic techniques and various algorithms that provide security and privacy in pervasive environments.

2.2 LIGHTWEIGHT CRYPTOGRAPHIC ALGORITHM

2.2.1 128-Bit Block Cipher CLEFIA

Shirai et al (2007) CLEFIA was developed jointly by Sony and the University of Nagoya. Similar to AES, it has a block length of 128 bits and offers three different key lengths: 128, 192 and 256 bits. CLEFIA uses 4-branch and 8-branch Type-2 generalized fiestel network and depending on the key length it takes 18, 22 or 26 rounds to encrypt one block of data.
Attacks

The designers of CLEFIA show resistance against differential, linear, impossible differential, saturation, algebraic and related key attacks. Their best attack can attack 10 of CLEFIA with 128 bit key, 11 rounds for 192 bit keys and 12 rounds for 256 bit keys. All attacks require more than $2^{100}$ chosen plain texts making them practically infeasible at the moment. For the attacks on 128 bit keys, however only $2^{32}$ blocks need to be stored. Also the attacks do not break the full-round CLEFIA and thus do not pose a threat.

2.2.2 PRESENT – An Ultra Lightweight Block Cipher

Bogdanov et al (2007) PRESENT is a 31 round cipher with substitution layer and permutation layer. The block size of PRESENT is 80 bit and key size is 80 bit. It was designed as an ultra lightweight block cipher. It is actually substitution permutation network. Each full round of PRESENT contains 4 layer as follows.

Initial transformation

Bitwise ex-or with round subkey s-box layer

Fixed 4 4-bit s-box is applied 16 times in parallel to intermediate cipher. Player

Fixed bit permutation

Output transformation

EX-OR with last round subkey
Differential and linear cryptanalysis

- Maximum differential probability of a present s-box is $2^{2}$
- Hence probability of a single $2^{5}$ round differential characteristic is bounded by $2^{100}$
- Data required to exploit the remaining $2^{5}$ round differential characteristic exceeds the amount available.

Structural attacks

Design of PRESENT is almost exclusively bitwise and permutation operation is somewhat regular. The development and propagation of word-wise structures are disrupted by bitwise operations used in the cipher.

Algebraic attacks

Increase in block size, addition of s-boxes and an appropriate version of linear diffusion layer yields in large system. Taking into consideration a system, which consists of 7 s-boxes of $2^8$ bits, it is difficult to get a solution in a reasonable time to 2-round version of the reduced cipher. Algebraic attacks are unlikely to pose a threat.

2.2.3 DES and DESXL

Enisa et al (2010) DESL and DESXL are two lightweight variants of Data Encryption Standard. The main idea is to simplify the DES round function using a single S-box instead of 8 and to discard the initial and final permutation of the DES to limit the size of the hardware implementation. So DESL iterates 16 rounds of a classical fiestel network and takes a block of size 64 bits as DES under a key of size 56 bits whereas DESXL uses, as
DESX, a whitening method to reinforce the security with a key of length 184 bits with 64 bit blocks. DESXL is considered for implementations for clear security reasons. No attack has been exhibited against DESL and DESXL. To store the block to cipher and the key, arrays of 8-bit numbers is used.

2.2.4 TEA

Mojtaba Alizadeh et al (2008) TEA is an old block cipher. TEA has been replaced by XTEA due to so many weakness. XTEA is a block cipher with 64 bits block length and 128 bits key length. It is based on fiestel network and recommended number of rounds is 64. Internal f-function composed of left and right shifts, XORs and additions. 3 subset of meet-in-the-middle attack is applied against 25 rounds of XTEA with 9 known plaintexts. KATAN and KTANTAN are two block ciphers based on stream cipher design. They both take as input blocks of sizes 32,48 or 64 bits under 80 bits key iterate during 254 rounds. KATAN and KTANTAN differ from their key schedules. In KATAN, 80 bits key is loaded into register whereas KTANTAN is fixed. Best attack against KATAN is conditional differential analysis. KATAN and KTANTAN make use of two round functions. The difference between those functions, is just adding xor where bit is 1. Therefore use bit field of „1“ when xor is applied and „0“ when it is not.

2.2.5 IDEA

Mickael Cazorla et al (2010) IDEA is a conventional block cipher with 64 bits block length and 128 bits key length. It is one of the widely used block ciphers, due to its inclusion in PGP cryptographic packages. IDEA is
resistant against very few attacks and the most significant attacks are 6 rounds attack which is faster than exhaustive key search.

2.2.6 SEA

Francois Xavier et al (2009) SEA is a lightweight block cipher that comprises of n bits block length and key length. It is very suitable block cipher as values 48, 96,144,etc., It is more efficient one to implement in hardware. It comprises of low memory size and small codes to be implemented. There is no security analysis published about SEA except one paper.

2.2.7 TWINE

Tomoyasu Suzaki et al (2011) TWINE is a lightweight block cipher with block length of 80 bits and whatever key length, it has 36 rounds. TWINE employs both encryption and decryption. It comprises of 1500GE area. Even though hardware size is small, it can be implemented in both hardware and software. Speed is more than other lightweight ciphers.

Tor Bjorstad et al (2010) author described about AES. AES is a block cipher with block length of 128 bits and key length of 128, 192 or 256 bits. AES is an iterated block cipher based on Substitution Permutation Network. It is composed of 10 rounds that repeat four elementary mappings such as SubBytes, ShiftRows, MixColumns and AddRoundKey. Plaintext is processed with non-linear, linear and key dependant information. Defensive mechanism against various security attacks.
Xinxin Fan et al (2010), describes an algorithm named Humming bird, which does not come under the category of block cipher or stream cipher, but it is having the property of both. It comprises of 256 key size and small block size, because it is suitable for sensors handling only small messages. One property of humming bird is updating the state after encryption. Authenticated encryption with associated data is a method in humming bird which authenticates any associated data that travels with cipher text. Processing of associated data happens only after entire encrypted payload has been processed. Without message expansion, communication of data is better for messages with size less than 16 bits. Plaintext is processed with initialization vector, X-OR operation and non-linear mixing function (mix the content of key register). Resist against differential cryptanalysis, cryptanalysis and related key attacks. To further reduce the consumption of the area and power of humming bird, four s-boxes used in Humming bird can be replaced by single s-box, which is repeated 4 times in 16-bit block cipher. In humming bird 2, power consumption is low and speed of processing is faster. Drawback of humming bird 2, control unit is more complicated and delay of critical path is much longer in Humming bird hardware than PRESENT.

Shirai et al (2007), proposed a new algorithm named CLEFIA which is a conventional block cipher and implemented in both hardware and software. It has block length of 128 bits and offers three different key lengths 128, 192 and 256 bits. Number of rounds such as 18, 22 or 26 depends on key length. CLEFIA uses 4-branch and 8-branch type-2 generalized fiestel network. Advantage is impossible differential cryptanalysis against CLEFIA. In expansion method, cryptanalysis is done for 13, 14 and 15 round CLEFIA.
Mojtaba Alizadeh et al (2010), decribes about TEA and KATAN. TEA is an old block cipher. TEA has been replaced by XTEA due to so many weakness. XTEA is a block cipher with 64 bits block length and 128 bits key length. It is based on fiestel network and recommended number of rounds is 64. Internal f-function composed of left and right shifts, XORs and additions. 3 subset of meet-in-the-middle attack is applied against 25 rounds of XTEA with 9 known plaintexts. KATAN and KTANTAN are two block ciphers based on stream cipher design. They both take as input blocks of sizes 32,48 or 64 bits under 80 bits key iterate during 254 rounds. KATAN and KTANTAN differ from their key schedules. In KATAN, 80 bits key is loaded into register whereas KTANTAN is fixed. Best attack against KATAN is conditional differential analysis. KATAN and KTANTAN make use of two round functions. The difference between those functions, is just adding xor where bit is 1. Therefore use bit field of ‘1’ when xor is applied and ‘0’ when it is not.

Mojtaba Alizadeh et al (2012), describes the LBLOCK ciphers, comprise block size of 64 bits under key size of 80 bits using 32 rounds in modified fiestel network. Round function is composed of subkey addition, 8 s-boxes in addition followed by 4 bits permutation. To prevent attack, modified key schedule algorithm is used, consists of complexity slightly lower than exhaustive key search. Also, zero correlation linear attack against LBLOCK which is managed to perform 22 rounds with a complexity.

Mickael Cazorla et al (2010), author explained about IDEA, SEA, TWINE and Piccolo. IDEA is a conventional block cipher with 64 bits blocks and 128 bits key. It is composed of 8.5 identical rounds. It is one of the widely used block ciphers, due to its inclusion in PGP cryptographic
packages. The most significant attacks are 6 rounds attack faster than exhaustive key search. SEA is a lightweight block cipher that comprises of n bits block length and key length. It is very suitable block cipher as values 48, 96, 144, etc.. It is based on modified two branches fiestel network. The f-function of fiestel is constructed using elementary operations and is composed of recommended number of rounds. There is no security analysis published about SEA except one paper.

TWINE is a lightweight cipher with block length of 80 bits and key length of 128 bits. Whatever key length, it has 36 rounds. TWINE employs fiestel structure with 16 branches. Internal f-function composed of subkey addition and single s-box, repeated 8 times per round.

Lightweight block ciphers are designed for various applications that resource constrained in nature. These ciphers are designed according to trade-off between size, cost and security. Several new and innovative designs were proposed depend on change in requirements. Cryptanalysis of block ciphers, optimal trade-off between various different parameters would be design.

2.3 COMPARISON TABLE

The Table 2.1 summarizes the comparison of light weight cryptographic ciphers covered in this survey. It brings out various differences in various cryptographic algorithms and gives us basic idea about which one is best compared to the others.
<table>
<thead>
<tr>
<th>Technology</th>
<th>Memory Size</th>
<th>Cost</th>
<th>RAM Occupancy</th>
<th>Cycles</th>
<th>Speed</th>
<th>Key size</th>
<th>Block size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Noekeon</td>
<td>Between 2000</td>
<td>High</td>
<td>34</td>
<td>&lt;5500 cycles</td>
<td>Slow</td>
<td>128</td>
<td>128</td>
</tr>
<tr>
<td>LED mCrypton</td>
<td>Between 2000</td>
<td>Low</td>
<td>41</td>
<td>&gt;5500</td>
<td>Slow</td>
<td>64</td>
<td>64</td>
</tr>
<tr>
<td></td>
<td>Between 2000 &amp; 3000 bytes</td>
<td>Low</td>
<td>&lt;20 bytes</td>
<td>&gt;5500</td>
<td>492.3</td>
<td>64</td>
<td>64</td>
</tr>
<tr>
<td>Piccolo</td>
<td>Between 2000 &amp; 3000 bytes</td>
<td>Low</td>
<td>91</td>
<td>&lt;5500</td>
<td>Slow</td>
<td>80/128</td>
<td>64</td>
</tr>
<tr>
<td>SEA</td>
<td>Between 2000</td>
<td>Low</td>
<td>24</td>
<td>&lt;5500</td>
<td>Slow</td>
<td>128</td>
<td>96</td>
</tr>
<tr>
<td>Klei</td>
<td>Between 4000</td>
<td>Low</td>
<td>36</td>
<td>&lt;5500</td>
<td>Slow</td>
<td>64, 80 or 96</td>
<td>64</td>
</tr>
<tr>
<td>TWIN</td>
<td>Between 2000</td>
<td>Low</td>
<td>23</td>
<td>&lt;5500</td>
<td>Fast</td>
<td>80 and 128</td>
<td>128</td>
</tr>
<tr>
<td>Katan &amp; Ktantan</td>
<td>Huge memory</td>
<td>Low</td>
<td>&gt;200</td>
<td>&gt;5500</td>
<td>25.1</td>
<td>80</td>
<td>64</td>
</tr>
<tr>
<td>Humming bird</td>
<td>Less memory</td>
<td>High</td>
<td>&lt;20 bytes</td>
<td>&lt;4000</td>
<td>400</td>
<td>128</td>
<td>16</td>
</tr>
<tr>
<td>HIGHT</td>
<td>Between 2000</td>
<td>Low</td>
<td>&lt;20 bytes</td>
<td>&lt;5500</td>
<td>188.2</td>
<td>128</td>
<td>64</td>
</tr>
<tr>
<td>LBlock</td>
<td>Between 2000</td>
<td>Low</td>
<td>&lt;20 bytes</td>
<td>&lt;5500</td>
<td>200</td>
<td>80</td>
<td>64</td>
</tr>
<tr>
<td>MIBS</td>
<td>Between 2000</td>
<td>Low</td>
<td>&lt;20 bytes</td>
<td>&lt;2000</td>
<td>Slow</td>
<td>80</td>
<td>64</td>
</tr>
<tr>
<td>SkipJack</td>
<td>Between 2000</td>
<td>Low</td>
<td>&lt;20 bytes</td>
<td>&lt;2000</td>
<td>Slow</td>
<td>80</td>
<td>64</td>
</tr>
<tr>
<td>TEA &amp; XTEA</td>
<td>&lt;1500</td>
<td>Low</td>
<td>&lt;20 bytes</td>
<td>&lt;2000</td>
<td>57.1</td>
<td>128</td>
<td>64</td>
</tr>
<tr>
<td>AES</td>
<td>&gt;3000</td>
<td>Low</td>
<td>&gt;20 bytes</td>
<td>&lt;2000</td>
<td>12.59</td>
<td>128</td>
<td>128</td>
</tr>
<tr>
<td>DESXL</td>
<td>&lt;1500</td>
<td>Nil</td>
<td>112</td>
<td>&lt;5500</td>
<td>44.4</td>
<td>64</td>
<td>64</td>
</tr>
<tr>
<td>Present</td>
<td>&lt;1500</td>
<td>Nil</td>
<td>142</td>
<td>&lt;5500</td>
<td>200</td>
<td>80</td>
<td>64</td>
</tr>
</tbody>
</table>
2.4 LIGHTWEIGHT CRYPTOGRAPHY

Lightweight cryptography (LWC) is a research that has developed in recent years and focuses in designing schemes for devices with constrained capabilities in power supply, connectivity, hardware and software. Schemes proposed include hardware designs, which are typically considered more suitable for ultra-constrained devices, as well as software and hybrid implementations for lightweight devices.

For establishing authentication and authorization in IoT, the use of lightweight cryptography is crucial. Constraints of low resource devices such as battery limitations require symmetric key cryptography to decrease power consumption of devices. Symmetric and asymmetric key cryptography can apply lightweight properties for trustworthy life-logging in IoT.

Lightweight cryptography (LWC) is a research that has developed in recent years and focuses in designing schemes for devices with constrained capabilities in power supply, connectivity, hardware and software. Schemes proposed include hardware designs, which are typically considered more suitable for ultra-constrained devices, as well as software and hybrid implementations for lightweight devices.

For establishing authentication and authorization in IoT, the use of lightweight cryptography is crucial. Constraints of low resource devices such as battery limitations require symmetric key cryptography to decrease power consumption of devices. Symmetric and asymmetric key cryptography can apply lightweight properties for trustworthy life-logging in IoT.
2.4.1 Symmetric Key Cryptography

Block ciphers with lightweight properties have been proposed for AES and DES, CLEFIA and Present.

Hash functions

SHA-3 do not satisfy lightweight requirements.

AES-128

AES is the current block cipher standard. It is the most used block cipher. AES is an iterated block cipher based on a SPN structure that ciphers block of size 128 bits under 128, 192 or 256 bits keys. We focus here on the case of AES-128 that cipher 128 bits blocks under a key length 128 bits. This AES version is composed of 10 rounds that repeat four elementary mappings (SubBytes, ShiftRows, MixColumns and AddRoundKey) on blocks seen as 4x4 byte matrices. The main security result against AES-128 is biclique cryptanalysis. It improves the key exhaustive search using particular relations linking together the keys through the key schedule and some bytes on internal state. The working of AES is shown in figure 2.1.
2.4.2 HIGHT

HIGHT is a lightweight block cipher that provides high security. It comprises of 64-bit block length and 128-bit key length. It is 8-bit processor oriented and operations are XOR, addition mod 28 and left bitwise rotation. It is hardware oriented rather than software oriented. 34 clock cycles are required to perform single encryption. Low cost, low power and ultra-light implementations could be done in HIGHT. 3048 gates are fixed in area 0.25\( \mu \)m. By optimizing the key schedule and control logic, the area is reduced to 2608 GE. Plaintext is processed with initial, round function and final
transformation. Constants generated increases the randomness of subkeys. HIGHT is resistant to slide attacks and related key attacks. The performance of HIGHT, shows memory efficient is better than SkipJack but worse than RC5. HIGHT requires reasonable memory size. HIGHT is better than RC5 in the aspect of power consumption. Processing speed is higher than other ciphers. The working of HIGHT algorithm is shown in figure 2.2.

![HIGHT Algorithm](image)

**Figure 2.2 HIGHT algorithm**

### 2.5 SECURITY ISSUES IN PERVERSIVE COMPUTING

Security in pervasive computing actually spreads over a broad area, and it consist of a large number of issues including the security during authentication, authorization, and access control. These issues are classified into several sections and it is being examined to provide a brief survey along with the present trend in each corresponding area. Almost all recent research related to security have been surveyed. The exits several works where agent-based applications have shown to be promising in achieving the goals. Some
other projects have come up that deal with different security issues related to Mobile Agent System. In order to prevent malicious use Farmer et al (1996) suggested that agents should communicate only with trusted and authenticated nodes. Hence several trust models appear which is discuss later. A scenario is described Rasmusson & Jansson (1996) where the credibility of a node will vary depending on the agents’ interaction with that node. Karjoth et al (1997) describes a method to defend against several types of attacks and to restrict an agent from occupying a specific resource for a long time. In a recent interesting study Eustice et al (2003) the researchers proposed a security model named ‘QED’ (Quarantine, Examination and Decontamination). QED was designed to provide several aspects of security which are well known for fixed infrastructures within the realm of a pervasive computing environment – virus scan, firewall, intrusion detection, and update and patch management. As part of an examination phase, the QED model incorporates a fixed infrastructure based security nodes which can provide updated virus scanners and patches. These nodes are seeking permission to enter in the network, and QED can push the nodes to receive the updated information as a precondition for entrance. The Quarantine phase performs the isolation of clients to ensure that they meet the local integrity constraints. On the other hand, the device can also decide not to access some of the available services of the network due to conflict with its own access policy. Clients are checked for potential vulnerabilities and malicious code in the Examination period. The probable investigations include virus scans and memory scans. During an active examination, clients need to go through all the defined investigations, whereas in passive investigation the system acknowledges a digital certificate that ensures that the corresponding client have passed similar checks in the previous environment. The Decontamina. The researchers in Bagci et al (2003) Schick et al (2005) have proposed a new architecture named Ubiquitous Mobile Agent System (UbiMAS) where they have shown some unique security characteristics. Here the agent contains all the personal
information and can request a service on behalf of the user. Finally, it actually accompanies the corresponding mobile user as he travels across different environments and domains. The system incorporates two types of agents, user agent and service agent. The system architecture operates in the following manner: A Message Delivery Engine deals with message delivery to and from agents. If the message is destined to an agent that resides in the current node then the message is passed to the PoBox. PoBox maintains a PoBox Adder and a queue for each agent connected with the agent node. PoBox Adder maintains the communication link between an agent and agent node. Using PoBox Adder and the queue, the system can ensure security in message transfer and protect itself from several attacks including DoS attack. The agent node can issue a timer that can dynamically change the length of the queue which facilitates the security features. If the Message Delivery Engine finds that the destined agent is not in the current node, it then forwards the message to the required node. An agent can migrate from one node to another using the migration engine and serializer or deserializer component. While sending messages from agents or nodes, the Message Delivery Engine performs packet formatting, inclusion of header, and other required security operations. Acknowledgement has been introduced to guarantee against packet loss. All the messages are sent in encrypted form. The security features enable the system to protect the agents from hosts as well as both agents and hosts from malicious agents. UbiMAS has been developed in Java as a service that runs on top of a middleware named Autonomic Middleware for Ubiquitous eNvironments (AMUN) Trumler et al (2004)

2.5.1 Authentication and Authorization

Authentication and authorization have long been discussed in pervasive computing. Both features are needed, in order to restrict any malicious user from entering the network or to prove one’s own identity.

Authors in Wullems et al (2004) present an authorization mechanism based on context awareness. Several other applications can cope with this mechanism without any difficulty as it incorporates GSSAPI (Generic Security Services Application Program Interface-RFC 2743) through the well known technology Kerberos. Here the authors used contextual information of different roles in generating a simpler access control policy. The whole architecture includes authentication service, authorization service, dynamic context services provided through dynamic context service manager (DCSM), dynamic context update mechanism and event update mechanism. Kerberos, LDAP and XML have been thoroughly used in building the authentication architecture. These open protocols increase the operability of the architecture over several platforms. This architecture can also function on LINUX. Kerberos protocol provides the cryptographic technique required for authentication and LDAP supports the required storage. Two types of roles, standard role and task, have been specified in the authorization service. Dynamic Context Service Manager (DCSM) deals with the responsibility of activating and deactivating these roles based on the contextual constraints. Dynamic Context Service (DCS) is responsible for collecting context data and providing the information to DCSM based on a predefined policy. This policy specifies the frequency for collecting contextual information, threshold values for various events that indicate when to apprize DCSM about those events, etc. Dynamic Context Update Mechanism modifies and updates a current contextual information object when it receives such a request from DCS or
authorization services as a result of activation/deactivation. In a research paper from Dartmouth College Minami & Kotz (2005) researchers present a context aware authorization mechanism based on rules and facts. This rule-based authorization differs from others in that it does not need any central server or certificate authority (CA) which will be trusted by all and will store all the contextual information. In this method, when a user wants to acquire a service from a resource or the server, that server issues a logical authentication query and sends it to the host of the resource. Each host has a knowledge domain with which it attempts to prove the authorization query. If it fails, it distributes several portions of the proof to multiple hosts.

Through this distribution, the computational overhead is actually reduced. After getting the sub-proofs from co-hosts, the host of the resource can declare the result of the query to be TRUE or FALSE, thus indicating grant or denial of access. By design, this approach facilitates confidentiality, integrity and scalability. In the architecture of the host, the ‘Query Handler’ deals with the remote request and ensure personal confidentiality rules. The ‘Query Issuer’ takes the responsibility of passing request for sub-proofs to other hosts and enforces personal integrity issues. The ‘Interference Engine’ attempts to compose the proof tree based on the rules and facts available in its knowledge domain. XProloga of Java has been used in building the prototype that evaluates the authorization query. In the Gaia Authentication Campbell et al (2002) the authors incorporates a number of authentication means where each authentication mechanism attains a specific value known as a ‘confidence value’.

This value ranges from 0 to 1 depending on the device and protocol used in the authentication process. In order to increase the confidence value, a specific authentication mechanism may include any number of authentication processes. Reasoning technique is used to formulate the net confidence value
from the partial confidence values. This authentication provides a unique feature which decouples the authentication procedures and authentication devices into two sections. The Authentication Mechanism Module (AMM) encompasses all the authentication procedures available like challenge-response, Kerberos, SESAME Kaijser & Parker (1994) etc. The Authentication Device Module (ADM) incorporates a module for each authentication device like PDA, smart badge, etc., and these modules are device dependent. This decoupling facilitates the incorporation of a new protocol in the AMM section or a new module in the ADM section for a new authentication device without interacting with the other section. In order to ensure lightweight CORBA services, universally Interoperable Core (UIC) has been used.

Style of authentication sometimes varies based on the situation. Authors in Bardram & Kjær (2003) have presented an authentication mechanism based on the proximity of the user. This protocol has been implemented in a hospital scenario using several components. In this context aware system, a Java Card has been used to contain the identity information of the user including an id, password and a pair of secret and public keys. In order to incorporate context awareness, the system encompasses context monitors and context servers.

The context aware infrastructure mainly provides location information. If both the identification and context aware system fail for some unknown reason, the entire system returns to the manual username/password system to ensure security. The authenticity of the entire system depends critically on the accuracy of the location identification. The authors also analyze several types of passive and active attacks and their impact on the system. In Aboudagga & Refaei (2005) authors have gone through several
authentication mechanisms and a number of categorizations of the systems have been provided based on different classification criterions.

2.5.2 Access Control

Many projects and frameworks have dealt with the mechanism of access control and related security issues. In 1996, a protocol named Policy Maker Blaze et al (1996) was implemented with options for setting policies and providing access right queries. Then a new model Role Based Access Control (RBAC) Lupu & Marriott (1995), Lupu & Sloman (1997) gained popularity that defined access based on the role of the user. Though this model tries to secure the system from unauthorized users based on this theme, sometimes it becomes very difficult to define roles for every user. Later several researchers proposed a central knowledge base for access control mechanism in their projects Chen & Finin (2004), Covington & Fogla (2002), Muhtadi et al (2003), In pervasive ad hoc scenarios, information are collected and stored in different ways through different devices in different environments. This becomes nearly impossible if the owner of the information has to grant separate access based on client, situation, category of information, etc. There have already been addressed to reduce the number of access right permissions like RBAC (Role Based Access Control), sharing of access right strategies over multiple domains, etc. Here the authors focus on information relationship and place this as a new axis for limiting the issue of access rights. The information relationship has been classified in to three categories: 1. Bundling based, 2. Combination based, and 3. Granularity based. Access rights are stored as SPKI/SDSI digital certificates Ellison & Frantz (1999) in the corresponding client rather than storing all access rights in a central server, thus ensuring the required distributed approach. Whenever a client receives an access right, it stores the right and corresponding information relationship. Later when the client seeks to access a different
resource, the stored access rights and information relationships are used to build a proof for that access, and permission will be granted if he succeeds in building the proof. Thus this methodology will reduce the interaction with the owner of the information in issuing access rights. The conditions needed for access rights have been formalized. Java has been used in building the framework. The facility of proving access rights have been incorporated in the framework provided by Howell & Kotz (2000).

Access right proofs have been built as Java classes. This protocol implementation has used the CSI (Contextual Service Interface) Judd & Steenkiste (2003) of the well-known Aura project Garlan et al (2003) as a test bed. If a user is denied access to any event, the user should receive specific feedback information based on the refusal. The pervasive computing environment involves thousands of scenarios as well as a dynamic access control policy that changes based on various contextual information such as role of the user, activity, and time. As a result, the same user might be initially granted access to a particular service and then be refused at other times thus causing confusion for the user. Consequently, merely showing a simple message ‘Access Denied’ is not enough from the users’ perspective. Being inspired by this scenario, some researcher in Urbana-Champaign have proposed a feedback model named ‘Know’ Kapadia et al (2004) that provides an optimal alternative solution. When access to a particular service is made available, it simultaneously ensures that system’s security and access control policy is not being disclosed. The feedback information certainly increases the usability and reliability of the system but there has to be a trade-off between quality and quantity of feedback and disclosure of access control policies.

In the pervasive computing environment, we need a security policy that will simultaneously be an unobtrusive mechanism to the user as well as
have the ability to discover the services available for the user in a transparent manner. The system needs a dynamic security policy which is flexible enough to update and modify on the fly. Both the user and the system need a secure access control and authorization mechanism that will act as a middleman and negotiate with both the parties to find a best possible service within the limitations imposed by both the participants. The augmentation of contexts in access control is enhancing the static security features towards dynamic security. Due to the distributed and ad hoc nature of the pervasive computing environment, this system is open to several unique vulnerabilities and suffers from quite a number of well-known problems whose reputed solutions are not applicable here. Along with this, the capability of pervasive devices varies widely in terms of memory storage, battery power, computational capability, etc. As a result, the only option left is to make the small, tiny pervasive devices more responsible for their own security. But the burden of the security features may be too large for them due to their limitations in battery power, memory storage and computational capability. In a recent work (Ranganathan 2004) author mentioned five obstacles in security and included barriers like privacy and trustworthiness of the devices as security issues.

- **Access control:** In case of access control, the system is based on the role and identity of the user. Again this privilege of accessing system resources and services is a variable which depends on the time, situation and other contextual information. Here the user needs to trust the pervasive computing environment including the resources and services available. At the same time the system needs to ensure the identity and access rights of the user. Though several access control mechanisms have been developed for several specific scenarios, we need a common framework which works in all scenarios with equal efficiency.
Privacy

In case of privacy two issues come up with equal priority:

1) Is privacy of the user being maintained?

2) Is privacy of the data being maintained?

Unlike distributed computing, pervasive computing likes to take user information and consider it as important contextual information. Though this contextual information plays a vital role in updating the system dynamically, it sometimes poses serious threat to the privacy of the user, especially in situations where people do not want to disclose their identity or location. In a research Muñoz et al (2003), Rodriguez et al (2003) in IMSS General Hospital, Mexico the researchers formulated an ad hoc contextual information based hospital system which was very useful from the perspective of doctors, nurses, resident doctors and other medical staffs within the boundary of the hospital. But the availability of user information and displaying information in public created a negative impact on some users. As a result, the researchers had to make some changes in the design and put abstraction in the private information Favela et al (2004), Tentori et al (2005). In Castro Valley, California, nurses of the medical center refused to incorporate the location tracking system as they believed that this would hamper their privacy (Reang 2002) As part of the famous Gaia project, developers have shown a privacy preserving hop by hop routing methodology Rahman & Hailes (1998) that carries information about the residing place of the user but it does not reveal the exact location or identity of the user. From these projects it is very evident that the privacy level and willingness of disclosure of personal information varies depending on information type, collection method, time and other concerns. In some scenarios users are reluctant to disclose identity information but don’t care much about location.
information. The situation might be reversed for some others, and there are scenarios where users are reluctant to reveal both. An intelligent system which can identify these issues and can dynamically adopt a mechanism in relation with other contextual variables can be a project of attention.

- Data communication

Privacy of the data encompasses two aspects. First, it has to ensure that data being shared or communicated is not being hacked by any active or passive eavesdroppers. As an initial thought, several encryption and decryption techniques is considered. But simultaneously we need to think about the other side of the coin which reminds us about the memory, battery power and other limitations.

Along with that, the users in pervasive computing environment have much more flexibility and independence in mobility. This includes a large variety of domains ranging from well secured environments to totally open unsecured situations which makes the data security issue worse. Secondly, how can it be guaranteed that the user data which is being collected almost transparently will not be used maliciously? Or how we can ensure with certainty that the sophisticated data is not being manipulated by any unauthorized user?

- Trust

In order to overcome several constraints, mutual cooperation, interconnectedness and inter dependability have been exposed as the obvious characteristics of pervasive computing environment. Along with these occurs the issue of trust. If data is shared with an unwarranted device, the probability of data security reduces automatically. Several trust models have been developed addressing various issues of trust Almenarez & Marin (2004),
Pirzada & McDonald (2004), Stajano & Anderson (1999). Again, what should be the exact criteria for proving trust? This is one of the questions yet to be resolved.

2.6 ANALYSIS OF SECURITY IN VARIOUS E-HEALTH NETWORKS

The authors Varsha Mary George et al (2014) have carried out a detailed survey on various security related issues in e-health care systems. Modern healthcare scenario is rapidly changing. The older generation is aging and their population is increasing. Additionally, diseases are becoming diverse and more severe in affecting both the elderly and children. Healthcare is turning out to be complex and expensive. For efficient and constant health monitoring, patients are fitted with sensors on their bodies which are either wearable or implantable, to form body area networks to monitor, record and transmit medical statistics for perusal by physicians. Sensors from different patients are networked or linked together to form health sensor networks. As the devices in these networks work on wireless technologies, their usual vulnerabilities are present in these networks as well.

The face of today's healthcare has changed from what it was more than a decade earlier. Physicians and health personnel are looking to technology more than ever to help in taking care of elderly patients, critical infants and others. The care of the elderly, for example those who are afflicted with a serious, incapacitating disease and need care 24/7 can be taken care of by making use of medical sensors which are either wearable or implantable. These sensors monitor and measure required medical readings and then transmit necessary medical information to a remote healthcare facility or centre which is connected to this network. These form what are called Body Area Networks (BANs) or Body Sensor Networks (BSNs) as shown in figure 2.3. The physician makes use of the data from these networks
to keep a constant eye on their patients remotely. This health setup comes very handy in emergency situations where a sudden change in a person's vital statistics might require immediate medical attention and instant communication without the least possible human error.

Biosensors are also employed in healthcare scenarios to the same effect mentioned before Almenarez et al (2004), Ko et al (2010). RFIDs are another element used in such networks for purposes of tracking and identification of patients and their medication routines Xiao et al (2006). The RFID tags are built into bracelets for example, and worn by patients inside the healthcare facility. The tag has an ID, which is associated with the patient and stored in a database to be read by the RFID reader. Health personnel can also have these RFID tags embedded in their cards to provide them with specific access and privileges when it comes to getting access to patients' rooms and their medical data. This involvement of electronics and information technology in healthcare is what gives rise to the term e-health. The medical data in such scenario can be stored in electronic form known in different circles by different names such as Electronic Patient Record (EPR) or Patient Information Record (PIR).

![Figure 2.3 Example scenario of wireless body area network](image)
Given all the good things about such electronic health networks, security is a major issue. This is especially true when critical medical data is transmitted, for instance when EPRs are transmitted through the network. There is a chance for malicious intruders to intercept the data or procure it through unlawful means, and then even manipulate it, which cause harm to the patient. This paper reviews various health networks and the security measures employed.

Every wireless network with sensors or other devices handling critical information is vulnerable to many security breaches. This is especially true when there is a wireless transmission channel to be secured against various attacks posed by malicious parties. The presence of patient health information records also needs to ensure the following security properties.

- Confidentiality makes sure that the information is understandable only to that person for whom it was meant.
- Integrity guarantees that no malicious alteration has been made to data while in transit or otherwise.
- Availability guarantees that data is handy to anyone who needs it at the right time.
- Authentication proves to the data receiver that it was indeed sent by the person who claimed to be the sender.
- Authorisation guarantees that only those who are authorised to perform certain actions can perform them.
- Self-organisation guarantees that the sensors are autonomous in handling sticky situations that arise in the network amongst them.
• Non-repudiation makes sure that no one can deny that a message was received or sent by them.

• Privacy & anonymity guarantees that no one can be targeted based on their identity or location, which is prevented from exposure Aldini et al (2009), Wang Attebury et al (2006).

2.6.1 Securing Data Exchange and Storage

Data collected in the health networks are private and critical information belonging to the patients, and should be accessible only to the patients and their respective medical personnel. The data can be stored in electronic form as Electronic Patient Record (EPR) or Patient Information Record (PIR). While passing through the network, these data must be secured from security breaches as they may contain highly sensitive data that patients want to keep private for various reasons, both personal and work related. It is also possible that malicious people who want to harm a person can access these records, if not secured properly.

Standard Encryption

Ko et al (2010) proposed MEDiSN, which is a wireless sensor network meant to measure patient vital signs. It carries out 128-bit Advanced Encryption Standard (AES) based symmetric encryption and decryption to protect the data gathered. This application falls behind when it comes to policy violations Ko et al(2010) though. Cherukuri et al (2003) used RC5 a symmetric key block cipher as the encryption algorithm for securing the communication in their biosensor network.

Quirino et al (2008) in regards to asymmetric encryption in WSNs in Quirino & Ribeiro (2008) talked about RSA public key algorithm (1024-bit RSA) being the most commonly used as it is standardized and has relatively
better efficiency. Elliptic curve cryptography is also discussed to have potential due to its higher energy efficiency and better performance, provided smaller keys are used. The authors also mention a new scheme that was proposed in 2008 called Multivariate Quadratic Quasigroup (MQQ). 160-bit MQQ which was compared in this paper showed to be magnitudes faster than RSA and ECC in hardware.

Xiao et al (2006) talked about the use of hash functions and pseudo random number generators for use in the encryption and re-encryption in the RFIDs. RSA and AES are some of the cryptographic algorithms used in RFIDs for encryption and decryption; lightweight versions are specially designed for use in them.

A. Using Algebraic Signature

Xiao et al (2006) proposed a secure data storage scheme in sensor networks. Initially when the data is stored, the original encrypted data is broken into n data shares, each of which consists of a data block generated from (n, k)-erasure coding, and a share of the secret key using (n, k)-secret sharing Xiao et al (2006). Then the shares of data are distributed to n neighbour nodes for storage. Here, any modification in data can be detected in the following manner. When a node wants to do an integrity check, all the other storage nodes compute and broadcast an algebraic signature on their data share, and the checking node verifies the integrity by checking its signature against those of other nodes.

B. Using Biometrics

Asymmetric cryptographic algorithm such as RC5 is used in Cherukuri et al (2003) to secure data exchange in the communication links among the biosensors and between the biosensors and the control node
outside. The data is encrypted using a suitable key. The key is generated based on biometric readings taken of the patient. These readings taken at different times in different situations result in different and truly random readings. This randomness is highly useful for the cryptographic randomness required when selecting keys. Poon et al (2006) made use of the variations in time found in heartbeat as a unique biometric characteristic for securing data. In Shanthini et al (2012), the author extracted the key from the fingerprint biometric trait, which was then randomized by applying a genetic operator.

2.6.2 Authentication

In Zhang et al (2009), static biometric traits were used to generate authentication keys and dynamic biometric traits to generate encryption keys. 64-bit and 128-bit keys were generated from electrocardiograms, photoplethysmograms and fingerprint images. The hamming distances between the keys were non-zero and their entropy was in the range from 0.662 to 1.

Without proper authentication anyone can access a patient's private data for malicious purposes. Authentication mechanisms are a necessary security measure to allow only authorised people access to data. Lu et al (2010) discussed about a mobile healthcare social network in which elderly patients can communicate with other elderly patients with similar symptoms. In order to prevent other people from knowing their symptoms and to identify those with the same symptoms, they have proposed a secure same symptom-based handshake (SSH) scheme. In the scheme, every patient is provided with a pseudo-ID and a private key corresponding to his/her symptom. So, if two patients meet, if they have the same symptom, they can use their respective private keys to mutually authentication each other. The scheme also explains about patient health information delivery through this connection among patients.
The authors in Liang et al (2012) proposed an attribute-oriented authentication scheme in. They explained about a health social network in which the users are each assigned attributes based on certain characteristics by the attribute trusted authority. Each of the HSN users have the ability to generate an attribute proof for themselves, whereby the sensitive attributes can be anonymized if they so choose. Only by verifying the provided attribute proof, as shown in figure 2.4, will the other users be able to know what attributes an HSN user has and thus can authenticate themselves to access others' personal medical data.

Recently, He et al (2013) proposed an authentication protocol for wireless medical sensor networks. It helps to authenticate the health professional in order to access a patient's physiological data. The protocol in He Kumar et al (2013) consists of professional registration and patient registration phases, followed by the login and authentication phase. The authentication is performed by inserting a smart card in a card reader, which inputs the professional's id and password, followed by using random numbers and checking timestamps for authenticating. This protocol provides user anonymity as well.

Althobaiti et al (2013)is an example of a biometric trait being used for authentication in wireless sensor network. They used the iris of the user to regenerate the user’s key every time the user needs be authenticated which considerably enhances the security. After extracting the iris’s features, the biometric encryption is performed by using a fuzzy commitment scheme, the biometric data is then stabilised. This stabilised data is bound with the user's random generated key during registration phase. The saved hash value of the encryption key is used during the remote authentication process.
2.6.3 Access Control and Privacy

Restricting access to a patient's private medical information data is an important facet of security.

Massacci et al (2009) discussed about accessing resources based on two things, one being that a person should be given rights explicitly to access that resource and second being that there should be a specific purpose for using it, hence their principle is stated as no purpose, no data. There are specific goals, roles and goals-roles assignments in this system based on which resources can be accessed; this access is provided by the access control manager. The predefined security policies in the system determines if there is any danger based on which emergency requests are sent to the medical centre.

In Mohan et al (2009) explained MedVault, a framework for sharing electronic health records. The EHRs are kept in a source verifiable repository. Identity agents and an authorisation module are also present. The request for a particular record from a user is met with compliance to associated access policies. The amount of information requested, even if by an authorised individual is considered so as to help protect a patient's privacy. This system provides fine-grained patient control over information disclosure thus safeguarding privacy, and data integrity and verifiability to both patients and
health care providers. The use of attributes facilitates role based access control as well.

Memon (2009) described role based access control in healthcare ad hoc networks. In RBAC, users are granted membership into roles based on their assigned responsibilities in the organization. Policies are defined for each group and role so the user is automatically assigned access rights to various data within the organisation. Each functional role has a set of permissions once it is activated for a particular session, and remains active for that session unless explicitly specified otherwise. RBAC is described as policy neutral and is said to support security policy objectives, and the static and dynamic separation of duty limitations which might come as a result of change in roles themselves or the policies associated with those roles.

2.7 RESEARCH MOTIVATION

In pervasive computing, devices are integrated into everyday objects and activities, and seamlessly communicate to share and exchange huge amounts of information. Technological innovation has enabled tiny devices to participate in pervasive computing; however, such devices are particularly vulnerable to security and privacy threats.

2.8 PROBLEM STATEMENT

The devices and technologies used in pervasive computing do not confine themselves well to data security. Pervasive computing suffers from very expensive operating costs, host bandwidths that are limited in nature and less data storage constraint.
2.9 RESEARCH OBJECTIVE

The objective of the research is to propose a novel framework for data security in order to provide high security and privacy for a pervasive e-healthcare system and integrate it as cloud based system so that the data storage constrain in handheld devices is overcome.