Attack Identification in a Cryptographic System using Dynamic Taint Propagation

Susan Basil, Panchami V

***Abstract*—Data is the critical resource of any organization. Cryptographic system protects the confidential data from intruders or attackers. Attackers make use of design flaws in existing algorithms to launch cryptographic attacks. Cryptography covers the areas like encryption, decryption and hashing for secure data transmission. In the proposed system, privacy sensitive information is tracked with the help of dynamic taint propagation. Active or passive attacks can be identified with the help of the taint propagation inspite of the existing cryptographic operations. Identification of the attacking node among the multiple nodes is made possible by making use of dynamic taint tracking. In a multimode system, attack intimation is given to the sender and receiver nodes when attack occurs at any of the intermediate nodes. Key matching at receiver node fails when an active attack occurs and results in avalanche effect. The proposed system identifies the attacker in a cryptographic system with the help of dynamic taint propagation.**

***Index Terms*— Attack Identification, Cryptography, Decryption, Encryption, Taint Propagation,**

I. INTRODUCTION

Data is the most important resource of any organization. Data leaking or unauthorized access of data may happen in the presence of an attacker. The data can be protected using cryptographic techniques like encryption/decryption and hashing. Chances of attack may still exist, even though, the data is protected. The attacker can be find out by tracking the data flow through the network. Taint propagation can be made use for the identification of the attacker.

A. Overview

Cryptography is the study of mathematical techniques related to aspects of information security such as confidentiality, data integrity, entity authentication and data authentication [1].

The main goals of Cryptography are [2]:

- *Privacy or confidentiality:* The security service used to protect the data from unauthorized disclosure.
- *Data Integrity:* Assurance that data received are exactly as sent by an authorized entity. That is, contain no modification, insertion, deletion, substitution or replay.
- *Authentication:* Assurance that the communicating entity is the one that it claims to be.
- *Access Control:* Prevention of unauthorized use of a resource.
- *Non-repudiation:* Protection against denial by any one of the entities involved in a communication.

B. Cryptographic Algorithms

Cryptographic algorithms are sequences of processes, or rules, used to encipher and decipher messages in a cryptographic system. There are many different types of cryptographic algorithms, though most of them fit into one of the two classifications symmetric and asymmetric. Symmetric algorithms, also known as symmetric-key or shared-key algorithms, work by the use of a key known only to the two authorized parties. While these can be implemented in the form of block ciphers or stream ciphers, the same key is used for both encrypting and decrypting the message [1].

The Data Encryption Standard (DES) and Advanced Encryption Standard (AES) are the most popular examples of symmetric cryptography algorithms. Asymmetric cryptography algorithms rely on a pair of keys- a public key and a private key. The public key can be revealed, but, to protect the data, the private key must be concealed [1]. Hash functions are functions that compress an input of arbitrary length to a result with a fixed length to protect the authenticity of information [3].

C. Taint Propagation

User inputs from untrusted sources are considered as tainted data. Identifying, tracking and preventing the improper use of such untrusted data is the domain of the taint problem. There are mainly two categories of approaches to attack taint problem. First, statically analyzing the code for the presence of taint vulnerabilities and second, dynamic approaches that track tainted data at runtime.

In order to track tainted user input, following need to be specified:

- *Sources:* A source is a method that returns user input. Usually these are methods that get HTML form input or read cookies stored on the client, or parse HTTP parameters. All strings emanating from sources must be marked tainted.
- *Propagation:* Strings from sources are usually manipulated to form other strings such as queries, or scripts, or file system paths. Strings that are derived from tainted strings also need to be marked tainted.
- *Sinks:* A sink is a method that consumes input or derivative of user input. This includes method that

Manuscript received April 8,2015
***Susan Basil**, PG Student, CSE, Toc H Institute of Science & Technology, Ernakulam, India.*
***Panchami V,** Assistant Professor, CSE, Toc H Institute of Science & Technology, Ernakulam, India.*

ISSN: 2278 – 7798

execute some form of code (such as script or SQL query), or methods that output data. Tainted strings must be prevented from being used as parameters to sinks [4].

D. Paper Organization

This paper is organized as follows. Section II overviews the past related work. Section III describes the problem definition. Section IV presents the proposed system. Section V concludes the paper.

II. Related Work

802.11 networks can operate in either ad-hoc mode or in infrastructure mode. Ad-hoc mode allows wireless devices to communicate directly with each other with no pre-existing network infrastructure. This ad-hoc network has no connection to the outside world. Infrastructure mode only allows devices to communicate with a central access point. The access point functions as an Ethernet bridge between the wireless media and a wired network [19].

Ad-hoc mode WLANs are normally less expensive to run, as no access points are needed for their communication. The main drawback of ad-hoc mode is that this topology cannot scale for larger networks and it lack some security features like MAC filtering and access control [20]. In [21], Dahill et al. proposed asymmetric cryptography for securing ad hoc routing protocols.

DES is the most widely used block cipher in world. It encrypts 64-bit data using 56-bit key. Use of DES has flourished especially in financial applications and is still standardized for legacy application use. There has been a considerable controversy over its security [5].

A cryptographic hash function is a hash function which is considered practically impossible to invert, that is, to recreate the input data from its hash value alone. These one-way hash functions have been called the workhorses of modern cryptography. The input data is often called the message, and the hash value is often called the message digest or simply the digest [6].

MD5 is the most widely used secure hash algorithm particularly in Internet-standard message authentication. The algorithm takes a message of arbitrary length as input and produces a 128-bit message digest as the output. This is mainly intended for digital signature applications where a large file must be compressed in a secure manner [7]. The existing system [9] that consists of multiple rounds of block cipher encryption and decryption followed by a final round of hash is a time consuming process.

User inputs from untrusted sources are considered as tainted data. Identifying, tracking and preventing the improper use of such untrusted data is the domain of the taint problem. There are mainly two categories of approaches to attack taint problem. First, statically analyzing the code for the presence of taint vulnerabilities and second, dynamic approaches that track tainted data at runtime [4]. According to B Livshits Runtime [8], taint tracking can be implemented at several levels, affecting the instrumentation precision, overhead, and level of implementation difficulty. It includes source-level instrumentation, bytecode-level instrumentation, library- level instrumentation, debugging APIs and runtime-level instrumentation.

The basic cause of security problems in source code can be find out using static analysis approaches. Errors can be find out in early development even before the program's initial run with the help of static analysis. Early error detection reduces the cost of error fixing and improves the developer coding approach. Various static analysis approaches have been proposed by researchers to detect vulnerabilities from the source code of software systems. Xin-Hua Zhang et al [11] implemented a taint analysis based tool to detect XSS attacks and SQL injection vulnerabilities. It tracks various kinds of external input, tags taint types, construct the control flow graph and taint data propagate to various kinds of vulnerability functions, but not free from the false positive problem [10].

J.Newsome and D.Song [12] proposed Dynamic Taint Analysis for Automatic Detection, Analysis, and Signature Generation of Exploits on Commodity Software. TaintCheck is a novel mechanism that uses dynamic taint analysis to detect a buffer overrun vulnerability or format string vulnerability. Manuel Egele, Christopher Kruegel and Engin Kirda [13] proposed a novel dynamic spyware analysis approach that precisely tracks the flow of sensitive information as it is processed by the web browser and any loaded browser helper objects (BHO). Dynamic taint analysis can be used to identify information flows and it tags and tracks sensitive data elements as they are processed.

III. Problem Definition

The avalanche effect can be used to pinpoint the cryptographic transient secrets from the execution of cipher data. The attacker should be find out and the data propagation to that particular node should be blocked if an attacker is present in the secure data transfer mechanism. If the attacker is able to correctly discover the plain data, then the intended receiver should not be permitted to decrypt the cipher data and the attack intimation should be send to the sender. For correct decryption of cipher data, the key matching is required.

Avalanche effect is used in order to uncover the cryptographic operations like encryption, decryption and hash from the execution of a given cipher data. Avalanche effect is the desirable property of all cryptographic algorithms like public key cryptographic algorithms, symmetric cryptographic algorithms and hash functions such that a slight change in the input would cause significant changes in the output. The proposed system can be used to detect the attacker in the system without false accusation using taint propagation mechanism. The presence of attacker is intimated to the sender if the receiver is not able to decrypt the cipher data.

ISSN: 2278 – 7798

IV. PROPOSED SYSTEM

This paper proposes a novel system for the secure communication of data and to detect the attacker without false accusation using cryptographic operations. The proposed system is a composition of encryption, hashing, taint propagation and decryption so as to find out the malware attack which can happen to the system, to uniquely identify the attacker and also to identify the cryptographic secrets in the system with the help of avalanche effect.Fig.1 shows the overview of the proposed system.

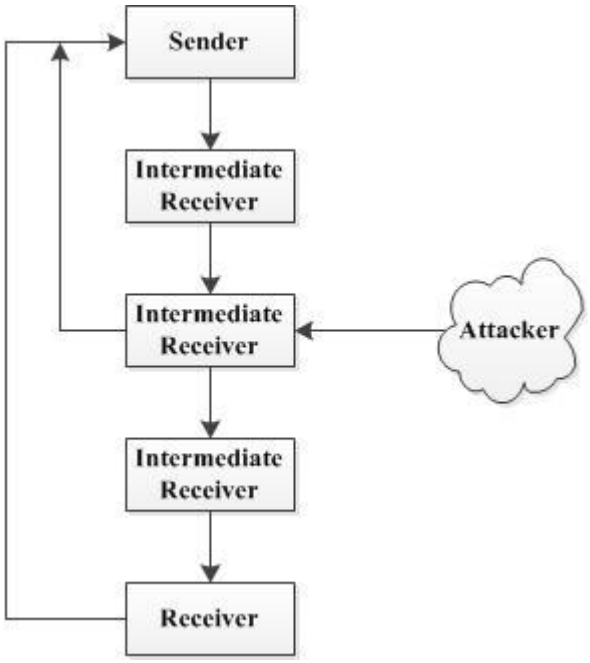

Fig 1. Overview of the Proposed System

Data after encryption and hashing passed to the intermediate node after tainting. Possibilities for attack are present at the intermediate node. The data on arriving at the receiver can identify whether an attack has occurred or not by checking for key matching and taint value during data decryption. Also the attacker node can be find out using taint mechanism.

The sender tags the encrypted and hashed input in the input buffer as taint sources. The taint sources which propagate to other memory bytes are tracked, and these bytes are called taint data. This taint data belongs to the category of sensitive data as it has undergone encryption and hashing. The taint source tagging module assigns tag to the taint source. Dynamic taint tracking and data flow monitoring are done by the taint propagation module.

The proposed system reveals the malware attack that take place between a sender and a receiver even after performing cryptographic operations on the input data. The sender encrypts, hash and taint the input data. The intermediate node can either behave as an attacker or a normal node. If the intermediate node tries to decrypt the cipher data or modify the plain text, taint tracking helps to find out the attacker node without false accusation. If there was no decryption at the intermediate nodes and the receiver could decrypt the data correctly by key matching, then the actual plain text is recovered indicating that no attack was there during data transmission. Failure in data decryption at the receiver result in avalanche effect due to key mismatch or unauthorized data decryption, and the information about the attack is intimated to the sender. There are mainly 4 modules.

They are:

- Connection Establishment between Sender and Receiver
- Data at the Sender
- Data at the Intermediate Receiver
- Data at the Destination Receiver

A. Connection Establishment between Sender and Receiver

Socket programming provides the communication mechanism between the two computers using TCP [14]. A socket is created by the client program for communication and tries to connect to the server using that socket. The server creates a socket object on its end when the connection is made for communication. By writing to and reading from the socket, the client and server can now communicate. The java.net.Socket class represents a socket, and the java.net.ServerSocket class provides a mechanism for the server program to listen for clients and establish connections with them [14].

The following steps occur when establishing a TCP connection between two computers using sockets [15]:

- The server instantiates a ServerSocket object, denoting which port number communication is to occur on.
- The server invokes the accept() method of the ServerSocket class. This method waits until a client connects to the server on the given port.
- After the server is waiting, a client instantiates a Socket object, specifying the server name and port number to connect to.
- The constructor of the Socket class attempts to connect the client to the specified server and port number. If communication is established, the client now has a Socket object capable of communicating with the server.
- On the server side, the accept() method returns a reference to a new socket on the server that is connected to the client's socket.

After the connections are established, communication can occur using I/O streams. Each socket has both an OutputStream and an InputStream. The client's OutputStream is connected to the server's InputStream, and the client's InputStream is connected to the server's OutputStream. TCP is a two-way communication protocol, so data can be sent across both streams at the same time [15].

Sender sends the selected file to the receiver after passing through a number of intermediate receiver nodes. Data is undergone DES encryption, MD5 hashing and tainting. Even though DES algorithm and MD5 hash is widely used in industrial applications, these algorithms are susceptible to attack. These attacks can be detected by using taint propagation.

ISSN: 2278 – 7798

B. Data at the Sender

At the sender side, the data file which is to be send to the destination is chosen and the data is encrypted using DES algorithm. Single encryption algorithm alone is used for encryption process. After encrypting the original data, the MD5 hash function is used and the message digest /hash value is generated. The generated secret key cannot be modified. Instead of sending the data to the destination receiver, the data is forwarded to the intermediate node after tainting. Tainting helps in tracing out the data propagation. Fig 2. shows the working of Sender node.

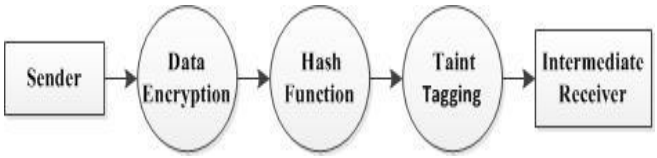

Fig 2. At the Sender

C. Data at the Intermediate Receiver

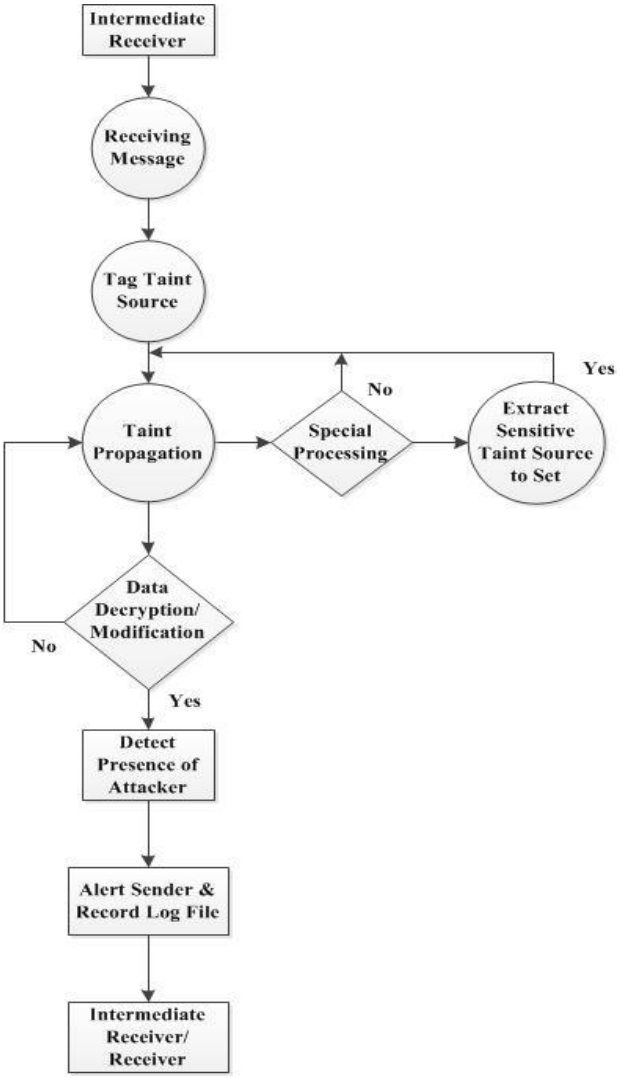

Fig 3. At the Intermediate Receiver

The intermediate node receives the message and tag the taint source. The tagged taint source is undergone taint propagation. In case of special processing, the sensitive taint source (i.e., after encryption and hashing) is extracted and is undergone taint propagation. If any data decryption is present, then it is considered as an attack and will alert the sender and is recorded in log file. The data is send to the next intermediate receiver or the receiver. Fig.3 shows the working of intermediate receiver node.

D. Data at the Destination Receiver

Finally the data will be reached to the destination receiver. This receiver will have the secret key to decrypt the data. If the key value at the receiver matches with that of the sender, and if no unauthorized data decryption took place in any of the intermediate nodes, the data will get decrypted successfully. If the data received is modified or if data decryption is detected during tainting, then an attack is intimated which result in avalanche effect. The taint propagation method is used to find the particular attacking node. If there is no data decryption or modification in any of the intermediate nodes, then the hash value is calculated using MD5 and data is decrypted successfully using DES. Fig.4 shows the working of the receiver node.

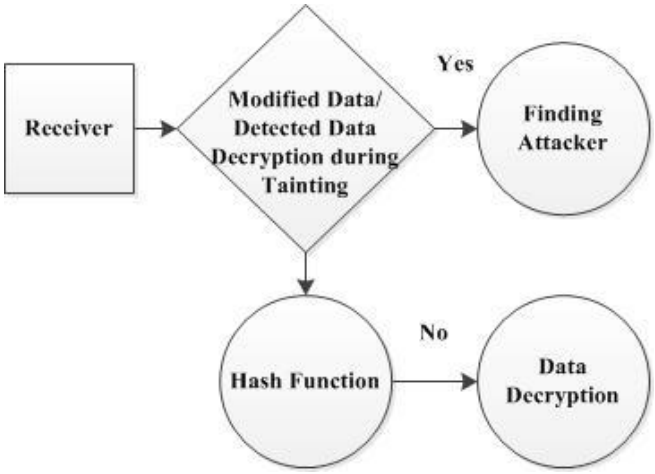

Fig 3. At the Receiver

V. Implementation And Analysis

The proposed system is implemented in Java. The sender is sending confidential data to the receiver via three intermediate nodes. The sender is having the option to encrypt the data with DES, generate hash with the help of MD-5 and a secret key is thus generated. The encrypted and hashed data is forwarded to the three intermediate nodes. Fig 4(a). shows the sender module. At the intermediate node-1, key is generated and data is forwarded to node-2. Similarly at node-2 and node-3.But there is a possibility of modification of data at all the intermediate nodes. If any such decryption/ modification happens at any of the intermediate nodes, it is considered as an attack. At the receiver, it is highlighted in red color and intimated that the data is corrupted as shown in Fig 4(e). If the data is not modified during its transmission, the actual sent data is recovered at receiver and the receiver is alerted saying *Data Reached Successfully* as shown in Fig 4(f).

ISSN: 2278 – 7798

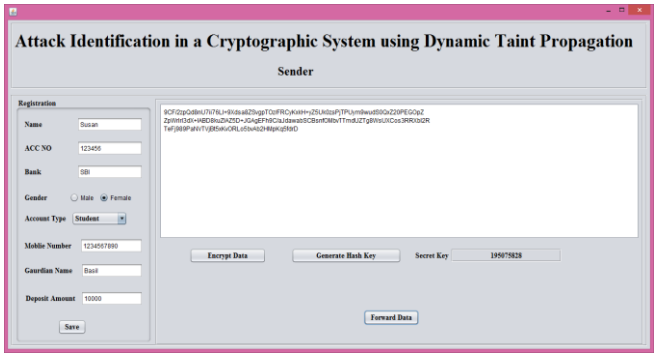

Fig 4(a). At the Sender

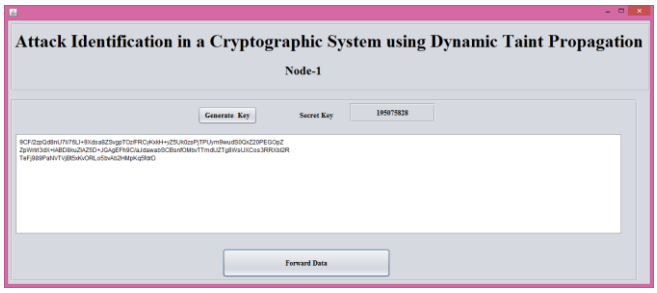

Fig 4(b). At Node-1

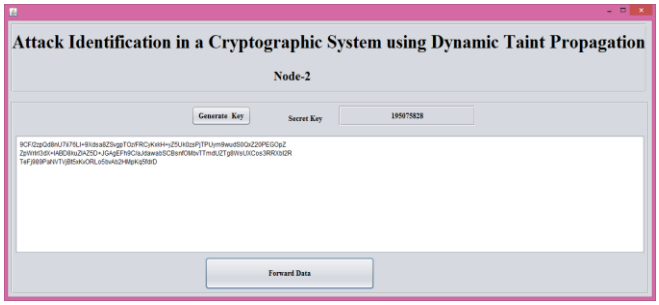

Fig 4(c). At-Node-2

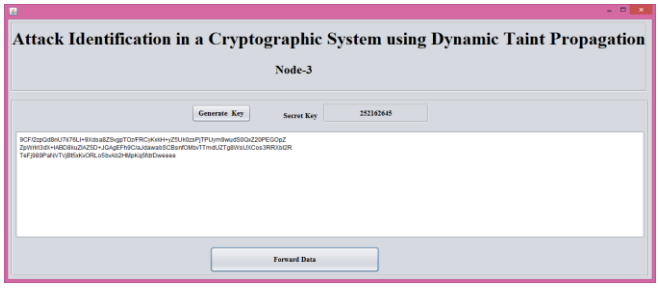

Fig 4(d). At-Node-3

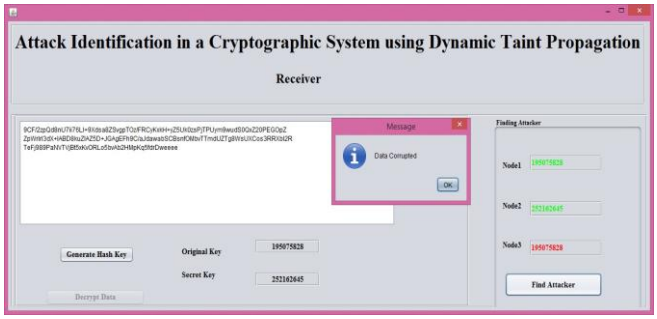

Fig 4(e).At Receiver-Attack Ocurred

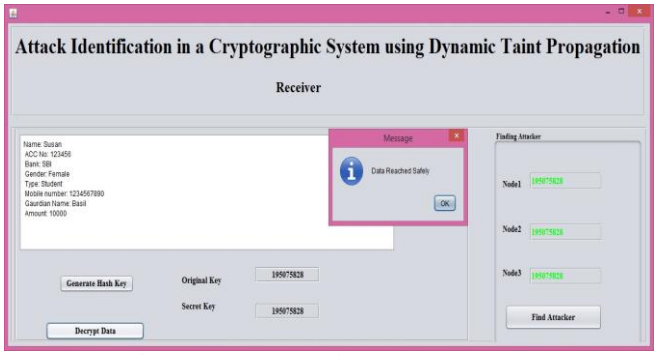

Fig 4(f).At Receiver-Successful Data Decryption

VI. CONCLUSION

A novel cryptographic system is presented which can address the security issues such as authentication, integrity, confidentiality, access control and non-repudiation that would detect the attacker with the help of taint propagation. Secure data transfer mechanism assured with the help of encryption and hashing can even undergo an attack and data can be modified by the attacker. If the intermediate receiver node succeeds in decrypting or modifying the cipher data, the intimation is send to the sender and the attacker node can be traced out through taint propagation. The data from intermediate receiver node is transmitted either to the next intermediate receiver or intended receiver. If the receiver is able to decrypt the cipher text sent by the intermediate node and there is no trace of decryption process in any of the intermediate nodes, then it is assumed that no attack has taken place during the propagation of the cipher data. If the receiver is not able to decrypt the cipher data correctly or if an unauthorized data decryption detected at any of the intermediate nodes, attack intimation is send to the sender. Taint propagation method is used to find out the attacker node. Avalanche effect results when the key matching technique fails at the receiver node.

REFERENCES

[1] Santosh Kumar Yadav, "Some Problems in Symmetric and Asymmetric Cryptography",A thesis submitted for the partial fulfillment of the degree of Doctor of Philosophy in Mathematics.
[2] William Stallings, "Cryptography and Network Security Principles and Practices Fourth Edition".
[3] Bart Preneel, "Analysis and Design of Cryptographic Hash Functions",February 2003.
[4] Vivek Haldar,Deepak Chandra and Michael Franz, "Dynamic Taint propagation for Java".
[5] Dr.Jean-Yves Chouinard, "Notes on the Data Encryption Standard(DES)", Design of secure Computer Systems,Sept23,2002.
[6] Gulshan Kumar, Anjala, Jyoti Sharma, "Authentication Techniques in Computer Networks",International Advanced Research Journal in Science, Engineering and Technology Vol.1, Issue 2, October 2014.
[7] Janaka Deepakumara, Howard M.Heys and R.Venkatesan, "FPGA Implementation Of MD5 Hash Algorithm".
[8] Benjamin Livshits, "Dynamic Taint Tracking in Managed Runtimes", Microsoft Research Technical Report.
[9] Xin Li,Xinyuan Wang and Wentao Chang, "CipherXRay: Exposing Cryptographic Operations and Transient Secrets from Monitored Binary Execution",IEEE Transactions On Dependable And Secure Computing, Vol.11, No.2, March/April 2014.
[10] Mukesh Kumar Gupta,Mahesh Chand Govil and Girdhari Singh , "An Approach to Minimize False Positive in SQLI Vulnerabilities Detection Techniques through Data Mining".
[11] Xin-Hua Zhang and Zhi-jian Wang, "A Static Analysis Tool for Detecting Web Application Injection Vulnerabilities for ASP

ISSN: 2278 – 7798

ProGram", 2nd International Conference on e-Business and Information System Security (EBISS), 22-23 May 2010.
[12] J.Newsome and D.Song, "Dynamic Taint Analysis for Automatic Detection, Analysis, and Signature Generation of Exploits on Commodity Software", Proc.12th Network and Distributed System Security Symp.(NDSS 05), Feb.2005.
[13] M.Egele, C.Kruegel, E.Kirda, H.Yin, and D.Song, "Dynamic Spyware Analysis", Proc.USENIX Ann.Technical Conf.(ATC 07), pp.233-246, June 2007.
[14] Rajat, Nitish Raj, R Harish, Shahbaz Ali Khan, Shokat Ali and Vaibhav Jain, "Networking with java (socket programming) a brief Study",Discovery Engineering, Volume 2, Number 7, October 2013.
[15] Hemant Kumar Srivastava, Rounak Sinha and Sumita Gupta, "Implementation of Socket Programming and RMI Using Simulating Environment",International Journal of Scientific & Engineering Research, Volume 4, Issue 5, May-2013.
[16] Weiming Li,Yilin Yan, Hao Tu , Jia Xu, "A Dynamic Taint Tracking Based Method to Detect Sensitive Information Leaking",IEICE-Asia-Pacific Network Operation and Management Symposium(APNOMS) 2014.
[17] Bellare, Mihir, Rogaway, Phillip, "Introduction to Modern Cryptography", p 10, 21 September 2005.
[18] A.Menezes, P.van Oorschot, S.Vanstone, "Handbook of Applied Cryptography", 1997.
[19] Lelia Barlow, "Security of Wireless Local Area Networks",June 2004.
[20] "Wireless Networking Security", Dec 2010©The Government of the Hong Kong Special Administrative Region.
[21] Kimaya Sanzgiri, Bridget Dahill, Brian Neil Levine, Clay Shields and Elizabeth M. Belding-Royer, "A Secure Routing Protocol for Ad Hoc Networks".
[22] Himani Agrawal and Monisha Sharma, "Implementation and analysis of various symmetric cryptosystems", Indian Journal of Science and Technology Vol.3 No.12, December 2010.

Susan Basil is a PG scholar in TIST under CUSAT. She was working as a Software Engineer in Wipro for two years. She completed BTech in IT from VJCET, Vazhakulam. She has presented a paper in international conference sponsored by IEEE at Greater Noida.

Panchami.V is a Research scholar in Anna University, Chennai. Now working as Asst.Professor in CSE department in TIST. She has 5 years of teaching experience. She is a B Tech. degree holder in IT and secured M Tech from Govt College of Engineering, Salem in CSE. She has presented technical papers in many national and international seminars and conferences. She had developed an Android application "ifollow"for ladies safety and got NASSCOM award for that app.

ISSN: 2278 – 7798