ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 5, May 2016

Secure Transmission of Provenance and Packet Drop Attack Detection in WSN with Light Weight Scheme

V Vaijayanthi [1], S.Geetha [2]

Pursuing M.Phil (Computer Science), Department of Computer Science, Shrimati Indira Gandhi College, Trichy, Tamil Nadu, India[1]

Assistant Professor, Department of Information Technology, Shrimati Indira Gandhi College, Trichy, Tamil Nadu, India[2]

ABSTRACT: Large-scale sensor networks are deployed in numerous application domains, and the data they collect are used in decision-making for critical infrastructures. Data are streamed from multiple sources through intermediate processing nodes that aggregate information. A malicious adversary may introduce additional nodes in the network or compromise existing ones. Therefore, assuring high data trustworthiness is crucial for correct decision-making. Data provenance represents a key factor in evaluating the trustworthiness of sensor data. Provenance management for sensor networks introduces several challenging requirements, such as low energy and bandwidth consumption, efficient storage and secure transmission. A novel lightweight scheme is proposed to securely transmit provenance for sensor data. The proposed technique relies on inpacket Bloom filters to encode provenance. Efficient mechanism is introduced for provenance verification and reconstruction at the base station. The secure provenance scheme is extended with functionality to detect packet drop attacks staged by malicious data forwarding nodes. The proposed technique is evaluated both analytically and empirically, and the results prove the effectiveness and efficiency of the lightweight secure provenance scheme in detecting packet forgery and loss attacks.

KEYWORDS: Provenance, Security, Sensor Networks, Light Weight Scheme, Packect Data Secure In WSN

I.INTRODUCTION

Sensor networks are used in numerous application domains, such as cyber physical infrastructure systems, environmental monitoring, power grids, etc. Data are produced at a large number of sensor node sources and processed in-network at intermediate hops on their way to a Base Station (BS) that performs decision-making. The diversity of data sources creates the need to assure the trustworthiness of data, such that only trustworthy information is considered in the decision process. Data provenance is an effective method to assess data trustworthiness, since it summarizes the history of ownership and the actions performed on the data.

Scope Of Work: In a multi-hop sensor network, data provenance allows the BS to trace the source and forwarding path of an individual data packet. Provenance must be recorded for each packet, but important challenges arise due to the tight storage, energy and bandwidth constraints of sensor nodes. Therefore, it is necessary to devise a light-weight provenance solution with low overhead. Furthermore, sensors often operate in an untrusted environment, where they may be subject to attacks. Hence, it is necessary to address security requirements such as confidentiality, integrity and freshness of provenance.

II.LITERATURE SURVEY

Provenance-Based Trustworthiness Assessment in Sensor Networks H. Lim, Y. Moon, and E. Bertino, Proc. Seventh Int'l Workshop Data Management for Sensor Networks, pp. 2-7, 2010 As sensor networks are being increasingly deployed in decision-making infrastructures such as battlefield monitoring systems and SCADA

(Supervisory Control and Data Acquisition) systems, making decision makers aware of the trustworthiness of the collected data is a crucial. To address this problem, we propose a systematic method for assessing the trustworthiness of data items. Our approach uses the data provenance as well as their values in computing trust scores, that is, quantitative measures of trustworthiness. To obtain trust scores, we propose a cyclic framework which well reflects the inter-dependency property: the trust score of the data affects the trust score of the network nodes that created and manipulated the data, and vice-versa. The trust scores of data items are computed from their value similarity and provenance similarity. The value similarity comes from the principle that "the more similar values for the same event, the higher the trust scores". The provenance similarity is based on the principle that "the more different data provenances with similar values, the higher the trust scores". **Chimera: A Virtual Data System for Representing, Querying, and Automating Data Derivation I. Foster, J. Vockler, M. Wilde, and Y. Zhao, Proc. Conf. Scientific and Statistical Database Management, pp. 37-46, 2002** A lot of scientific data is not obtained from measurements but rather derived from other data by the application of computational procedures. We hypothesize that explicit representation of these procedures can enable documentation of data provenance, discovery of available methods, and on-demand data generation (so-called "virtual data"). To explore this idea, we have developed the Chimera virtual data system, which combines a virtual data catalog for representing data derivation procedures and derived data, with a virtual data language interpreter that translates user requests into data definition and query operations on the database. We couple the Chimera system with distributed "data grid" services to enable on-demand execution of computation schedules constructed from database queries. We have applied this system to two challenge problems, the reconstruction of simulated collision event data from a high-energy physics experiment, and searching digital sky survey data for galactic clusters, with promising results. **A Survey of Data Provenance in E-Science Y. Simmhan, B. Plale, and D. Gannon, ACMSIGMODRecord, vol. 34, pp. 31-36, 2005.** Data management is growing in complexity as large-scale applications take advantage of the loosely coupled resources brought together by grid middleware and by abundant storage capacity. Metadata describing the data products used in and generated by these applications is essential to disambiguate the data and enable reuse. Data provenance, one kind of metadata, pertains to the derivation history of a data product starting from its original sources.In this paper we create a taxonomy of data provenance characteristics and apply it to current research efforts in e-science, focusing primarily on scientific workflow approaches. The main aspect of our taxonomy categorizes provenance systems based on why they record provenance, what they describe, how they represent and store provenance, and ways to disseminate it. The survey culminates with an identification of open research problems in the field. **The Case of the Fake Picasso: Preventing History Forgery with Secure Provenance R. Hasan, R. Sion, and M. Winslett, Proc. Seventh Conf. File and Storage Technologies (FAST), pp. 1-14, 2009.** As increasing amounts of valuable information are produced and persist digitally, the ability to determine the origin of data becomes important. In science, medicine, commerce, and government, data provenance tracking is essential for rights protection, regulatory compliance, management of intelligence and medical data, and authentication of information as it flows through workplace tasks. In this paper, we show how to provide strong integrity and confidentiality assurances for data provenance information. We describe our provenance-aware system prototype that implements provenance tracking of data writes at the application layer, which makes it extremely easy to deploy.

III.MODULES

Network Model

A multihop wireless sensor network, consisting of a number of sensor nodes and a base station that collects data from the network. The network is modeled as a graph G(N,L), where N ={ni|, 1<=i<=|N|} is the set of nodes, and L is the set of links, containing an element $l_{i,j}$ for each pair of nodes ni and nj that are communicating directly with each other. Sensor nodes are stationary after deployment, but routing paths may change over time, e.g., due to node failure. Each node reports its neighboring (i.e., one hop) node information to the BS after deployment. The BS assigns each node a unique identifier nodeID and a symmetric cryptographic key Ki. In addition, a set of hash functions H ={h1, h2, . . . , hk} are broadcast to the nodes for use during provenance embedding.

Aggregated Data

Assume a multiple-round process of data collection. Each sensor generates data periodically, and individual values are aggregated towards the BS using any existing hierarchical (i.e., tree-based) dissemination scheme. A data path of D hops is represented as <nl, n1, n2, . . ., nD >, where nl is a leaf node representing the data source, and node ni is i hops away from nl. Each non-leaf node in the path aggregates the received data and provenance with its own locally-generated data and provenance.

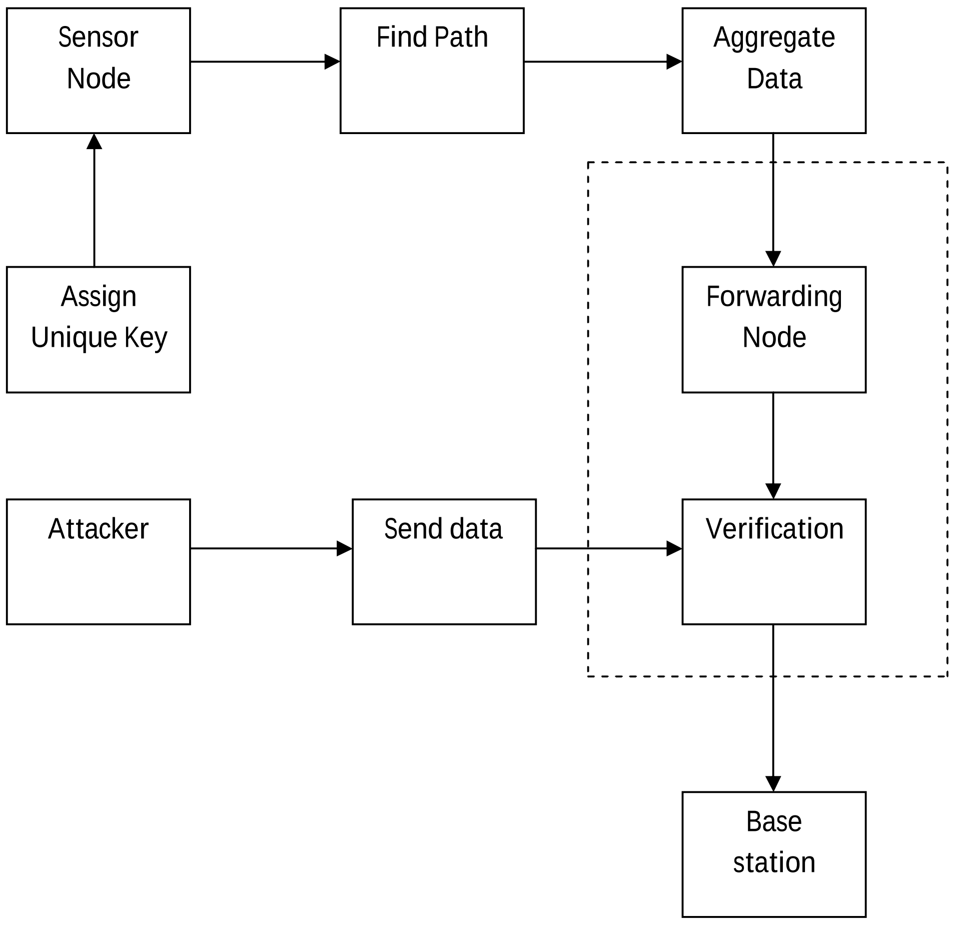
Provenance Encoding

Consider node-level provenance, which encodes the nodes at each step of data processing. This representation has been used in previous research for trust management. For a data packet, provenance encoding refers to generating the vertices in the provenance graph and inserting them into the iBF. Each vertex originates at a node in the data path and represents the provenance record of the host node. A vertex is uniquely identified by the vertex ID. The VID is generated per-packet based on the packet sequence number (seq) and the secret key Ki of the host node. We use a block cipher function to produce this VID in a secure manner.

Provenance Verification And Decoding

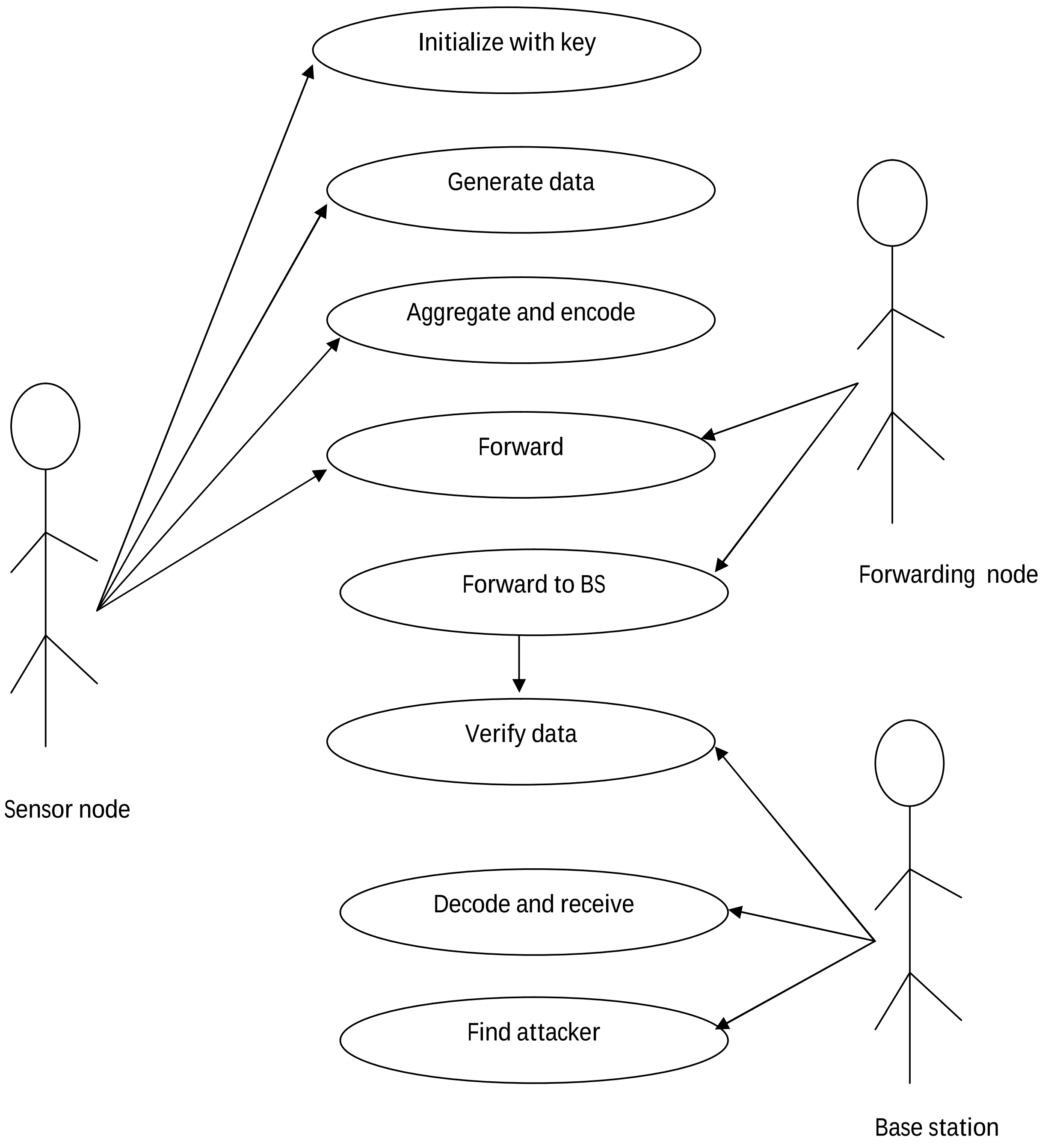
When the BS receives a data packet, it executes the provenance verification process, which assumes that the BS knows what the data path should be, and checks the iBF to see whether the correct path has been followed. However, right after network deployment, the path of a packet sent by a source may not be known to the BS. In this case, a provenance collection process is necessary, which retrieves provenance from the received iBF and thus the BS learns the data path from a source node. Afterwards, upon receiving a packet, it is sufficient for the BS to verify its knowledge of provenance with that encoded in the packet.

CHART 1 : SYSTEM ARCHITECTURE



IV.UML DIAGRAM

CHART 2: USE CASE DIAGRAM



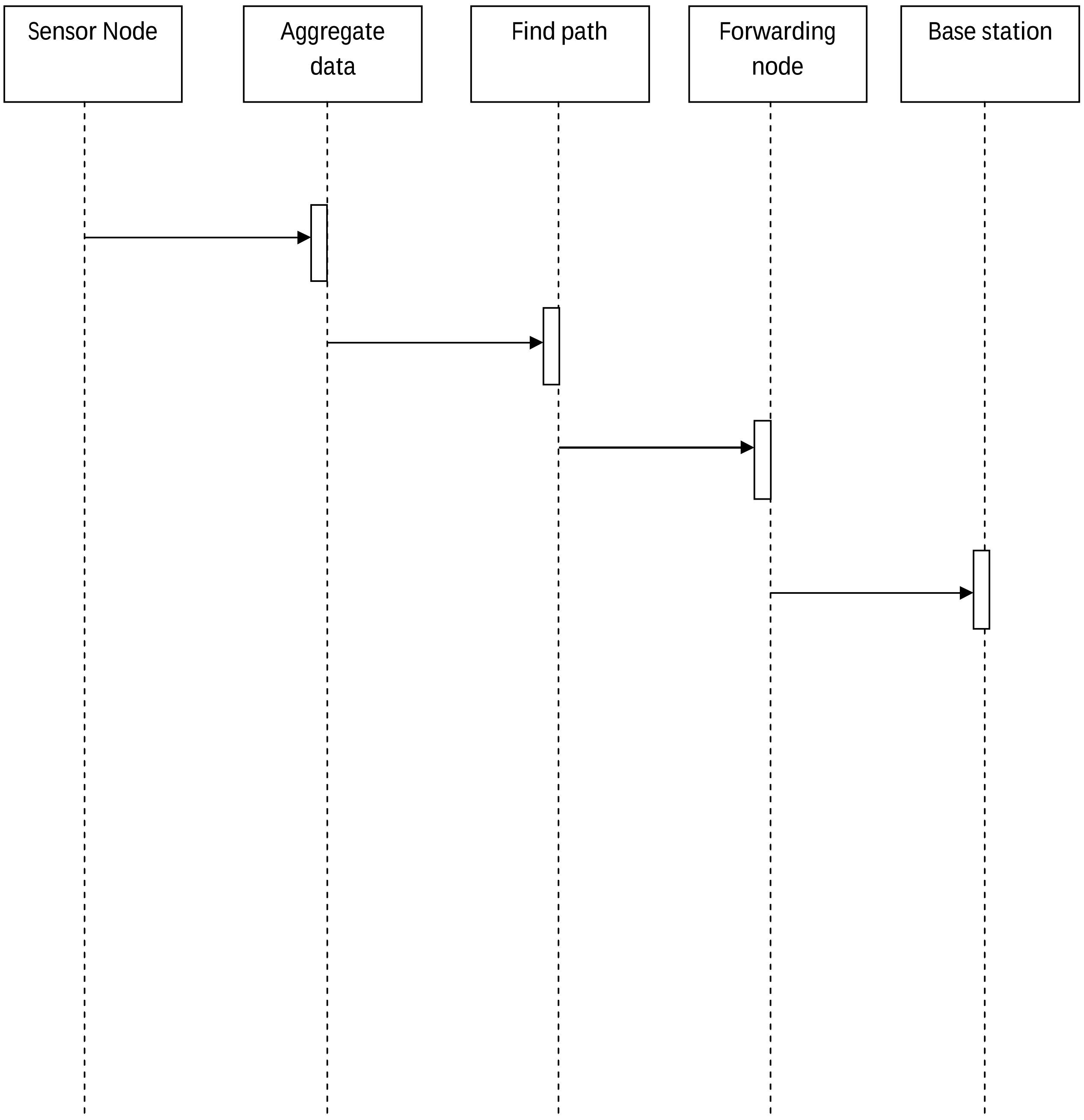
ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 5, May 2016

CHART 3: SEQUENCE DIAGRAM



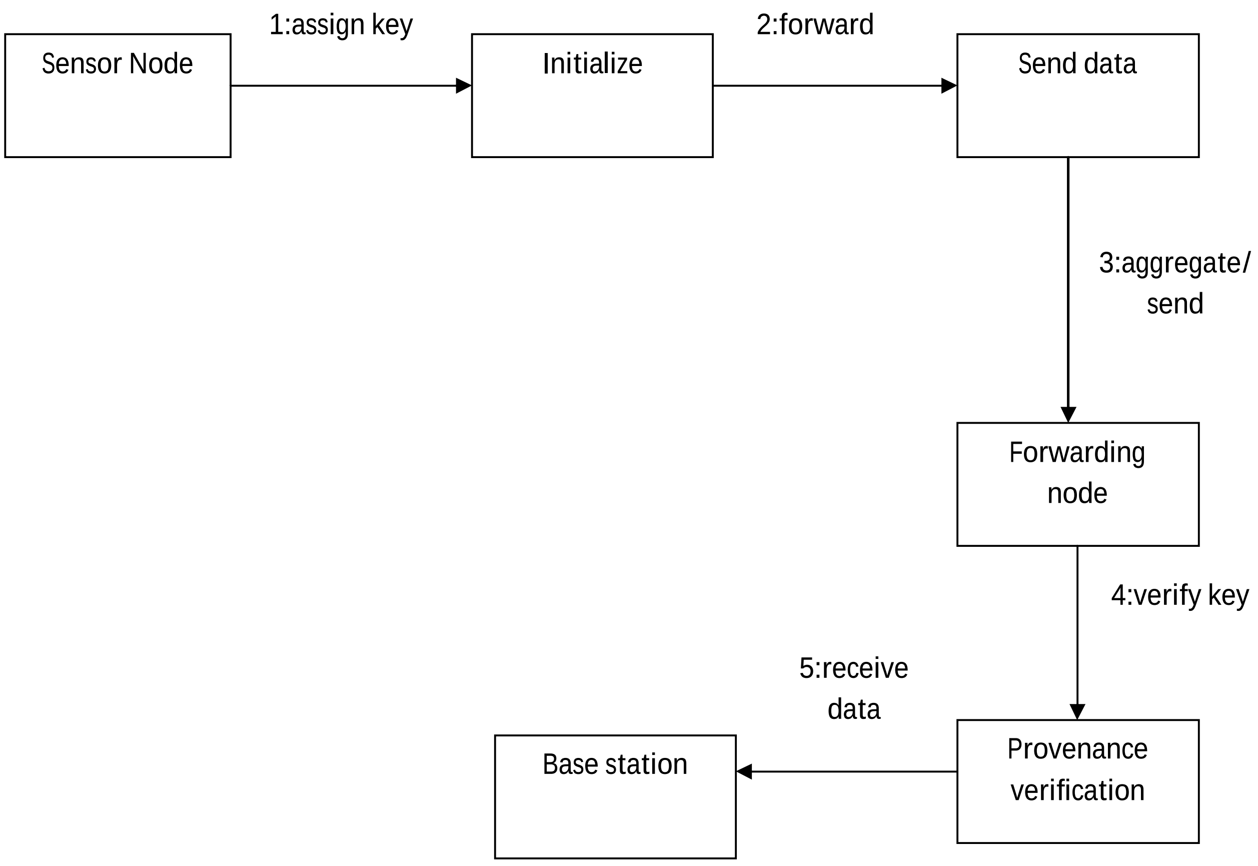
ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 5, May 2016

CHART 3: COLLABORATION DIAGRAM



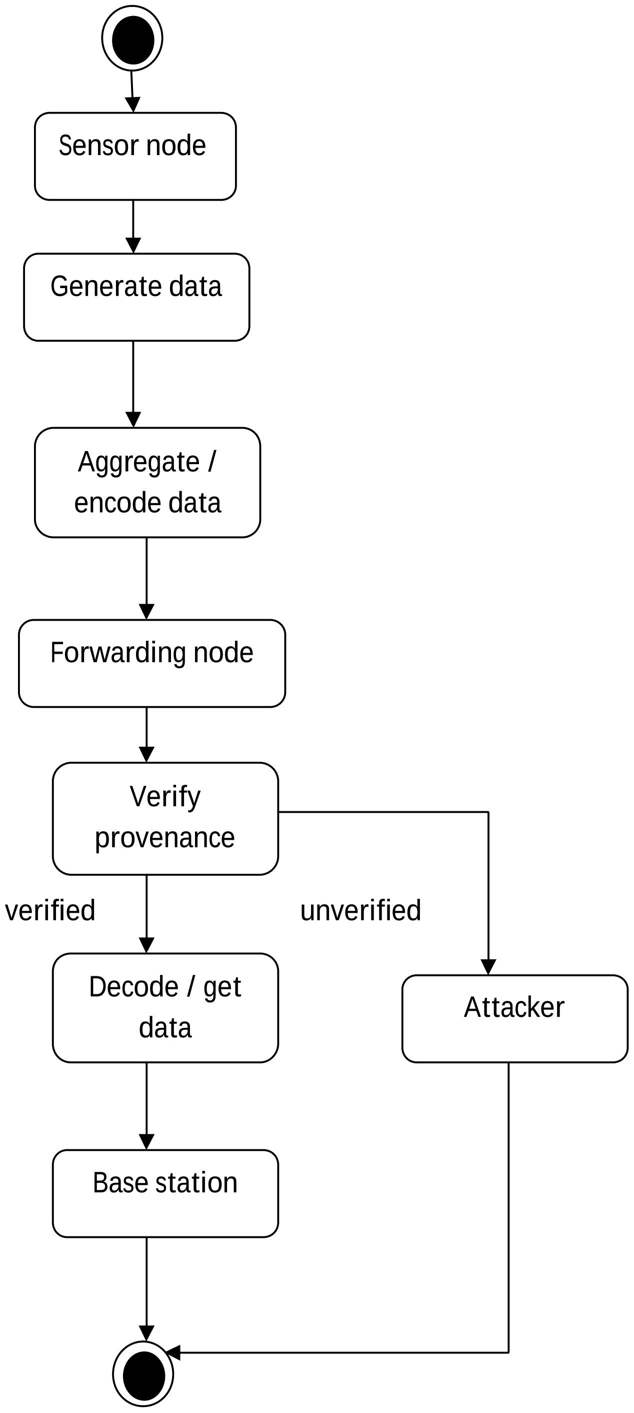
ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 5, May 2016

CHART 4: ACTIVITY DIAGRAM



ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 5, May 2016

V.TESTING

Introduction

After finishing the development of any computer based system the next complicated time consuming process is system testing. During the time of testing only the development company can know that, how far the user requirements have been met out, and so on.

Software testing is an important element of the software quality assurance and represents the ultimate review of specification, design and coding. The increasing feasibility of software as a system and the cost associated with the software failures are motivated forces for well planned through testing.

Testing Objectives

These are several rules that can save as testing objectives they are:

- Testing is a process of executing program with the intent of finding an error.
- A good test case is one that has a high probability of finding an undiscovered error.

Testing procedures for the project is done in the following sequence

- System testing is done for checking the server name of the machines being connected between the customer and executive..
- The product information provided by the company to the executive is tested against the validation with the centralized data store.
- System testing is also done for checking the executive availability to connected to the server.
- The server name authentication is checked and availability to the customer
- Proper communication chat line viability is tested and made the chat system function properly.
- Mail functions are tested against the user concurrency and customer mail
- date validate.

Following are the some of the testing methods applied to this effective project:

SOURCE CODE TESTING

This examines the logic of the system. If we are getting the output that is required by the user, then we can say that the logic is perfect.

SPECIFICATION TESTING:

We can set with, what program should do and how it should perform under various condition. This testing is a comparative study of evolution of system performance and system requirements.

MODULE LEVEL TESTING:

In this the error will be found at each individual module, it encourages the programmer to find and rectify the errors without affecting the other modules.

UNIT TESTING:

Unit testing focuses on verifying the effort on the smallest unit of software-module. The local data structure is examined to ensure that the date stored temporarily maintains its integrity during all steps in the algorithm's execution. Boundary conditions are tested to ensure that the module operates properly at boundaries established to limit or restrict processing.

INTEGRATION TESTING:

Data can be tested across an interface. One module can have an inadvertent, adverse effect on the other. **Integration testing** is a systematic technique for constructing a program structure while conducting tests to uncover errors associated with interring.

VALIDATION TESTING:

It begins after the integration testing is successfully assembled. Validation succeeds when the software functions in a manner that can be reasonably accepted by the client. In this the majority of the validation is done during the data entry operation where there is a maximum possibility of entering wrong data. Other validation will be performed in all process where correct details and data should be entered to get the required results.

RECOVERY TESTING:

Recovery Testing is a system that forces the software to fail in variety of ways and verifies that the recovery is properly performed. If recovery is automatic, re-initialization, and data recovery are each evaluated for correctness.

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 5, May 2016

SECURITY TESTING:
Security testing attempts to verify that protection mechanism built into system will in fact protect it from improper penetration. The tester may attempt to acquire password through external clerical means, may attack the system with custom software design to break down any defenses to others, and may purposely cause errors.
PERFORMANCE TESTING:
Performance Testing is used to test runtime performance of software within the context of an integrated system. Performance test are often coupled with stress testing and require both software instrumentation.
BLACKBOX TESTING:
Black- box testing focuses on functional requirement of software. It enables to derive ets of input conditions that will fully exercise all functional requirements for a program. Black box testing attempts to find error in the following category: Incorrect or missing function, Interface errors, Errors in data structures or external database access and performance errors.
OUTPUT TESTING:
After performing the validation testing, the next step is output testing of the proposed system since no system would be termed as useful until it does produce the required output in the specified format. **Output format** is considered in two ways, the **screen format** and the **printer format.**
USER ACCEPTANCE TESTING:
User Acceptance Testing is the key factor for the success of any system. The system under consideration is tested for user acceptance by constantly keeping in touch with prospective system users at the time of developing and making changes whenever required.

TABLE 1: TEST CASES

Sl. No	Test Case Name	Test Procedure	Pre-Condition	Expected Result	Passed/ failed
1	Neighbor detection	Create base station only	No other nodes are created	No neighbor is detected	Passed
2	Neighbor detection	Create base station and one forwarding node only	No other nodes are created	Only one neighbor is detected	Passed
3	Neighbor detection	Create base station and two forwarding node only	No other nodes are created	No sensor nodes are detected	Passed
4	Neighbor detection	Create base station and two forwarding nodes and three sensor	No other nodes are created	Sensor nodes are detected as neighbors	Passed
5	Path detection	Click on search button in sensor node	No other button clicked	Path is generated	Passed
6	Data sensing	Click on start button in sensor node	No other button clicked	Sensor node starts data sensing	Passed
7	Data sending	Click on start button in sensor node	No other button clicked	Sensor node send data to Cluster Head	Passed
8	Data aggregation	Click on start button in sensor node	No other button clicked	Sensor data aggregated in base station	Passed

ALGORITHM: Provenance verification and time verification
Input : Received packets with Seq ID , IBF, time ‘T’, Set of hash function H, data path ‘P’

```
BF_c <--- 0;
        For each n1 Є P|
                Vid = generate VID(SEQID, n|)
                Register Initiation time 'T|'
                Insert vid, Time into BF_c using hash function H
        Endfor
        If (BF_c = ibf) and (T=T|)
        Return true
        Endif
        Return false
```

VI.PROPOSED SYSTEM

The problem of secure and efficient provenance transmission and processing for sensor networks is discussed. Provenance is used to detect packet loss attacks staged by malicious sensor nodes. A provenance encoding strategy is proposed whereby each node on the path of a data packet securely embeds provenance information within a Bloom filter that is transmitted along with the data. Upon receiving the packet, the BS extracts and verifies the provenance information. To devise an extension of the provenance encoding scheme that allows the BS to detect if a packet drop attack was staged by a malicious node. Only fast Message Authentication Code (MAC) schemes and Bloom filters (BF) is used, which are fixed-size data structures that compactly represent provenance. Bloom filters make efficient usage of bandwidth, and they yield low error rates in practice.

Advantages

Secure provenance transmission in sensor networks, and identify the challenges specific to this context; Proposed an in-packet Bloom filter provenance encoding scheme; Efficient techniques for provenance decoding and verification at the base station;Detects packet drop attacks staged by malicious forwarding sensor nodes.

VII.CONCLUSION

Addressed the problem of securely transmitting provenance for sensor networks, and proposed a light-weight provenance encoding and decoding scheme based on Bloom filters. The scheme ensures confidentiality, integrity and freshness of provenance. The scheme extended to incorporate data-provenance binding, and to include packet sequence information that supports detection of packet loss attacks. Proposed scheme is effective, light-weight and scalable. In future work, we plan to implement a real system prototype of secure provenance scheme, and to improve the accuracy of packet loss detection, especially in the case of multiple consecutive malicious sensor nodes.

REFERENCES

[1] H. Lim, Y. Moon, and E. Bertino, "Provenance-Based Trustworthiness Assessment in Sensor Networks," Proc. Seventh Int'l Workshop Data Management for Sensor Networks, pp. 2-7, 2010.
[2] I. Foster, J. Vockler, M. Wilde, and Y. Zhao, "Chimera: A Virtual Data System for Representing, Querying, and Automating Data Derivation," Proc. Conf. Scientific and Statistical Database Management, pp. 37-46, 2002.
[3] K. Muniswamy-Reddy, D. Holland, U. Braun, and M. Seltzer, "Provenance-Aware Storage systems," Proc. USENIX Ann. Technical Conf., pp. 4-4, 2006.
[4] Y. Simmhan, B. Plale, and D. Gannon, "A Survey of Data Provenance in E-Science," ACMSIGMODRecord, vol. 34, pp. 31-36, 2005.
[5] R. Hasan, R. Sion, and M. Winslett, "The Case of the Fake Picasso: Preventing History Forgery with Secure Provenance," Proc. Seventh Conf. File and Storage Technologies (FAST), pp. 1-14, 2009.
[6] S. Madden, J. Franklin, J. Hellerstein, and W. Hong, "TAG: A Tiny Aggregation Service for Ad-Hoc Sensor Networks," ACM SIGOPS Operating Systems Rev., vol. 36, no. SI, pp. 131-146, Dec. 2002.
[7] K. Dasgupta, K. Kalpakis, and P. Namjoshi, "An Efficient Clustering Based Heuristic for Data Gathering and Aggregation in Sensor Networks," Proc. Wireless Comm. and Networking Conf., pp. 1948- 1953, 2003.
[8] S. Sultana, E. Bertino, and M. Shehab, "A Provenance Based Mechanism to Identify Malicious Packet Dropping Adversaries in Sensor Networks," Proc. Int'l Conf. Distributed Computing Systems (ICDCS) Workshops, pp. 332-338, 2011.
[9] L. Fan, P. Cao, J. Almeida, and A.Z. Broder, "Summary Cache: A Scalable Wide-Area Web Cache Sharing Protocol," IEEE/ACM Trans. Networking, vol. 8, no. 3, pp. 281-293, June 2000.