Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IMPACT FACTOR: 6.199

IJCSMC, Vol. 8, Issue. 5, May 2019, pg.203 – 207

A Novel Protocol for Two-Tiered Sensor Networks for Preserving Privacy and Integrity

Archana N[1]

[1]Department of Computer Science,
KSSEM,VTU, India
[1] archana05narayan@gmail.com

Megha J[2]

[2]Department of Computer Science,
KSSEM,VTU, India
[2] megha.j@kssem.edu.in

Abstract- In many wireless sensor network applications, a Sink needs to collect all the data that are sensed by the sensors which are located at different places. Since, sensors are memory limited readings from sensor nodes are aggregated at intermediate nodes to reduce the communication cost. In this paper we are making use of two-tiered sensor network architecture, where storage nodes serve as an intermediate tier between sensors and a sink for storing data and processing queries. It has been widely adopted because of the benefits of power and storage saving for sensors as well as the efficiency of query processing.

In this paper, we propose SafeQ, a protocol that helps to protect the data which is moving between sensor nodes to Data storage or Data storage to Sink or Sink to Data storage and Sink to user. It prevents attackers from gaining information from both sensor collected data and sink issued queries. SafeQ also allows a sink to detect the corrupted file if it is modified. To preserve privacy, SafeQ uses a novel technique to encrypt both data and queries such that a storage node can correctly process encrypted queries over encrypted data without knowing their values. To preserve integrity, a hash message authentication code (HMAC) is generated at the Data storage for verifying the integrity of the query result. So that sink can use this information to verify whether the result of a query contains exactly the data items that satisfy the query.

Keywords—Data storage, Encryption, Integrity, Privacy, SafeQ, WSN

I. Introduction

Wireless sensor networks (WSNs) have been widely deployed for various applications, such as environment sensing, building safety monitoring, earthquake predication, etc. In this paper, we consider a two-tiered sensor network architecture in which storage nodes gather data from nearby sensors and answer queries from the sink of the network. The storage nodes serve as an intermediate tier between the sensors and the sink for storing data and processing queries. Storage nodes bring three main benefits to sensor networks. First, sensors save power by sending all collected data to their closest storage node instead of sending them to the sink through long routes. Second, sensors can be memory-limited because data are mainly stored on storage nodes.

Third, query processing becomes more efficient because the sink only communicates with storage nodes for queries. However, the inclusion of storage nodes also brings significant security challenges. As storage nodes store data received from sensors and serve as an important role for answering queries, they are more vulnerable to be compromised, especially in a hostile environment. A compromised storage node imposes significant threats to a sensor network. First, the attacker may obtain sensitive data that has been, or will be, stored in the storage node. Second, the compromised storage node may return forged data for a query. Third, this storage node may not include all data items that satisfy the query.

The architecture of two-tiered sensor networks, where storage nodes serve as an intermediate tier between sensors and a sink for storing data and processing queries, has been widely adopted because of the benefits of power and storage saving for sensors as well as the efficiency of query processing. However, the importance of storage nodes also makes them attractive to attackers. In our project, we propose SafeQ, a protocol that prevents attackers from gaining information from both sensor collected data and sink issued queries, which typically can be modelled as range queries, and allows the sink to detect compromised storage nodes when they misbehave. For privacy, compromising a storage node should not allow the attacker to obtain the sensitive information that has been, and will be, stored in the node, as well as the queries that the storage node has received, and will receive. Note that we treat the queries from the sink as confidential because such queries may leak critical information about query issuers' interests, which need to be protected especially in military applications.

For integrity, the sink needs to detect whether a query result from a storage node includes forged data items or does not include all the data that satisfy the query. There are two key challenges in solving the privacy and integrity-preserving range query problem. First, a storage node needs to correctly process encoded queries over encoded data without knowing their actual values. Second, a sink needs to verify that the result of a query contains all the data items that satisfy the query and does not contain any forged data.

➢ A SafeQ protocol

In this paper, we propose SafeQ, a novel privacy- and integrity-preserving range query protocol for two-tiered sensor networks. The ideas of SafeQ are fundamentally different from the S&L scheme. To preserve privacy, SafeQ uses a novel technique to encode both data and queries such that a storage node can correctly process encoded queries over encoded data without knowing their actual values. It allows all the data to be transmitted in encrypted format within the system by using AES algorithm. To preserve integrity, it generates a message authentication code(MAC) for the result by using HMAC-MD5 algorithm and use this as a integrity verification information such that a sink can use this information to verify whether the result of the query contains all the data that satisfy the query and does not include any forged data. In this project, all three sensors are designed to collect only the temperature values.

➢ Problem Statement

The main aim of this paper is to design the storage scheme and the query protocol in a privacy and integrity preserving manner in a two-tiered sensor networks. Here we first check the data and query privacy. Data privacy means that a storage node cannot know the actual values of sensor collected data. This ensures that an attacker cannot understand the data stored on a compromised storage node. Query privacy means that a storage node cannot know the actual value of sink issued queries. This ensures that an attacker cannot understand, or deduce useful information from, the queries that a compromised storage node receives. Then we check the data integrity. This means if a query result that a storage node sends to the sink includes forged data or excludes legitimate data, the query result is guaranteed to be detected by the sink as invalid.

➢ Scope and Objective

This paper is designed with three sensor nodes which are located at three different places sensing the temperature of the particular area. The main scope here is to encrypt the sensor collected temperature data and the sink issued a query for preserving the data and query privacy and data integrity.

The main objectives of this paper are as follows:

- To provide significantly better security and privacy.
- To prevent the attackers from gaining information from the sensor collected data and sink issued queries.
- To allow a sink to detect the modified or corrupted file/data from the data storage.
- To encode both data and queries to preserve privacy.
- To generate integrity verification information to preserve the data integrity.
- To answers the queries from the user efficiently.

II. LITERATURE REVIEW

B. Sheng and Q. Li [1]proposed a scheme to preserve the privacy and integrity of range queries in sensor networks . This scheme uses the bucket-partitioning idea for database privacy. The basic idea is to divide the domain of data values into multiple buckets, the size of which is computed based on the distribution of data values and the location of sensors. In each time-slot, a sensor collects data items from the environment, places them into buckets, encrypts them together in each bucket, and then sends each encrypted bucket along with its bucket ID to a nearby storage node. For each bucket that has no data items, the sensor sends an encoding number, which can be used by the sink to verify that the bucket is empty, to a nearby storage node. When the sink wants to perform a range query, it finds the smallest set of bucket IDs that contains the range in the query, and then sends the set as the query to storage nodes. Upon receiving the bucket IDs, the storage node returns the corresponding encrypted data in all those buckets. The sink can then decrypt the encrypted buckets and verify the integrity using encoding numbers. The S&L scheme only considered one-dimensional data and it can be extended to handle multidimensional data by dividing the domain of each dimension into multiple buckets. The S&L scheme has two main drawbacks inherited from the bucket-partitioning technique. First, the bucket-partitioning technique allows compromised storage nodes to obtain a reasonable estimation on the actual value of both data items and queries. Second, for multidimensional data, the power consumption of both sensors and storage nodes, as well as the space consumption of storage nodes, increases exponentially with the number of dimensions due to the exponential increase of the number of buckets [1].

H. Hacigumus, B. Iyer and S. Mehrotra introduce a database-service-provider model for executing SQL over encrypted data, where user data resides on the premises of the database-service provider. Most corporations view their data as a valuable asset. The service provider would need to provide sufficient security measures to guard data privacy. At least two data-privacy challenges arise. The first challenge is: how do service providers protect themselves from theft of customer data from hackers that break into their site and scan disks? Encryption of stored data is the straightforward solution. The second challenge is that of total data privacy, which is more complex since it includes protection from the database provider. The requirement is that encrypted data may not be decrypted at the provider site. A straightforward approach is to transmit the requisite encrypted tables from the server (at the provider site) to the client, decrypt the tables, and execute the query at the client[2].

D. Boneh and B. Waters , proposed a public-key system for supporting conjunctive, subset, and range queries on encrypted data. In this public-key system a secret key can produce tokens for testing any supported query predicate. The token lets anyone test the predicate on a given cipher text without learning any other information about the plaintext. It provides a general framework for analyzing security of searching on encrypted data systems[3].

M. Narasimha and G. Tsudik proposed a scheme which focuses on verifying the completeness of the result of relational database queries. Merkle hash trees have been used for the authentication of data elements, and they were used for verifying the integrity of database queries. It proposed similar schemes for verifying the integrity of relational database query results using signature aggregation and chaining. For each tuple in a database, we computed the signature of the tuple by signing the concatenation of the digests of the tuple itself as well as the tuple's left and right neighbors. This scheme computed the signature by signing the concatenation of the digests of the tuple and its left neighbors along each dimension. Although our neighborhood chaining technique seems similar to the above signature aggregation and chaining technique, it is much more efficient and suitable for sensor networks. First, our technique concatenates a data item with its left neighbor without computing their digests. Second, our technique does not compute signatures, which require the use of computationally expensive public key cryptography [4].

III.SYSTEM ARCHITECTURE

This provides a high level overview of how the functionality and responsibilities of the system were partitioned and then assigned to subsystems or components. The main purpose here is to gain a general understanding of how and why the system was decomposed and how the individual parts work together to provide the desired functionality.

The following Figure 1 depicts the architecture of proposed system. A two-tired sensor network consists of three types of nodes: sensors, storage nodes (data server), and a sink.

- **Sensors:** Sensors are inexpensive sensing devices with limited storage and computing power. They are often massively distributed in a field for collecting physical or environmental data, e.g., temperature.

- **Data Storage:** Data storage/Storage nodes are powerful wireless devices that are equipped with much more storage capacity and computing power than sensors. Each sensor periodically sends collected data to its nearby storage node.

- **Sink:** The sink is the point of contact for users of the sensor network. Each time the sink receives a question from a user, it first translates the question into multiple queries and then disseminates the queries to the corresponding storage nodes, which process the queries based on their data and return the query results to the sink. The sink unifies the query results from multiple storage nodes into the final answer and sends it back to the user.

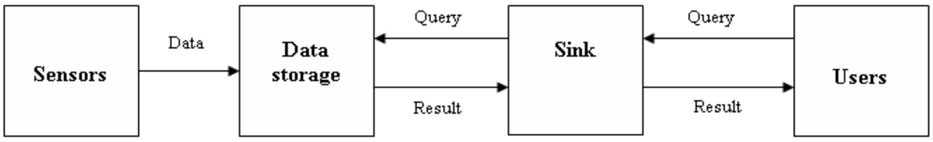

Figure 1 System Architecture

IV. METHODOLOGY

➢ Major Modules

The implementation of this paper has 4 modules namely:

- Sensor Module
- Login Module
- User Module
- Sink Module
- Data Storage Module

Each Module work according to fulfill the requirement of paper. The implementation of each module is conducted in such a way that it should not show any errors or exceptions. This implementation also tested with integrating several modules for the expected result.

A) Sensor Module

This is a simple module in which each sensor which is placed in different areas once activated, starts collecting the temperature values randomly. It uses the random value function for generating the temperature values. It then sends the encrypted temperature values to the data storage.

B) Login Module

This is a simple module in which user can enter with a valid user id and password. If user enters an incorrect user id or password it will display an error message i.e. invalid user id or password.

C) User Module

- ➢ After the successful login of the user, he/she can select the particular query types.
- ➢ Based on the query selected, he/she can select the particular date, time and also areas if necessary.
- ➢ This module then sends the selected query to the Sink.
- ➢ User can also view the computed results i.e. sent by the Sink.

D) Sink Module

- ➢ This module gets the user request from the login module; it then encrypts this request to preserve privacy.
- ➢ It also gets the selected query from the user module and encrypts the query then sends it to the data storage.
- ➢ It gets the query result from the data storage, if mac verification is successful sends the result to the user.

E) Data Storage Module

- This module collects/stores the encrypted temperature values along with the date, time and area from the Sensor module.
- It also gets the encrypted query from the Sink module.
- It then processes the encrypted query over the encrypted data and then sends the encrypted results to the Sink for verification.

V. CONCLUSIONS

In this paper, we proposed SafeQ protocol that helps in moving the temperature data and user queries safely from Sensor nodes to the Data storage and Data storage to the Sink respectively. All the data are in encrypted format so that it prevents attackers from gaining information from both sensor collected data and sink issued queries. SafeQ first preserves the data and query privacy and then preserves the data integrity. To achieve this, SafeQ uses a novel technique to encode both data and queries and generates integrity verification information so that a sink can use this information to verify the result of a query. Hence in this paper both the data and the queries moves safely from sensors to data storage and sink to data storage respectively.

REFERENCES

[1] B. Sheng and Q. Li, “Verifiable privacy-preserving range query in two tiered sensor networks,” in *Proc. IEEE INFOCOM*, 2008, pp. 46–50.

[2] H. Hacigümüş, B. Iyer, C. Li, and S. Mehrotra, “Executing SQL over encrypted data in the database-service-provider model,” in *Proc. ACM SIGMOD*, 2002, pp. 216–227.

[3] D. Boneh and B. Waters, “Conjunctive, subset, and range queries on encrypted data,” in *Proc. TCC*, 2007, pp. 535–554.

[4] M. Narasimha and G. Tsudik, “Authentication of outsourced databases using signature aggregation and chaining,” in *Proc. DASFAA*, 2006, pp. 420–436.