Maintaining Privacy and Integrity of Range Queries in Wireless Sensor Networks

R. Swathi[1], A. L. Sreenivasulu[2]

[1]PG student, Department of CSE, Intell Engineering College, Anantapur, AP, India

[2]Assistant Professor, Department of CSE, Intell Engineering College, Anantapur, AP, India

Abstract: ***The Two-tiered sensor network model has been widely adopted because of its storage and energy saving benefits, where storage node serves as an intermediate tier between sensors and sink for storing and processing queries. Storage node may contain vital data; we should recognize that they are vulnerable to attack. Therefore, we assume that storage nodes are not trustworthy but the sink is completely trustworthy. Here we propose SafeQ, protocols that prevents attackers from gaining information from both sensor collected data and sink issued queries .A 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 values to privacy. We have two schemes to preserve integrity, they are- one using Merkle hash trees and another using a new data structure called neighborhood chains to generate integrity verification information so that sink can verify whether the result of the query contains exactly the data items that satisfy the query. For improving performance, we have an optimization technique using bloom filters to reduce the communication cost between sensors and storage nodes.***

Keywords: Privacy, Integrity, SafeQ, Sensors, Storage nodes

1. Introduction

A wireless sensor network (wsn) in its simplest form can be defined as a network of (possibly low-size and low-complex) devices denoted as nodes that can sense the environment and communication the information gathered from the monitored field through wireless links; the data is forwarded, possibly via multiple hops relaying, to a sink that can use it locally, or is connected to other networks (e.g., the Internet) through a gateway.

We shall consider two-tiered wireless sensor network with three main actors. Sensors are in charge of sensing data. The sink receives queries from users, contacts the inner network to get answers, and returns them to users. Storage nodes stores data from sensors and seek answers for queries from the sink. The makes query processing more efficient because the sink needs to contact a few storage nodes instead of all the sensors. The storage nodes also brings many security challenges since the storage node stores the data received from sensors and serve as an important role for answering queries, they are more vulnerable to be compromised, mainly in a hostile environment. A compromised storage node causes threats to a sensor network. In this paper, we solve the problem in a two-tiered wsn of un-trustworthy storage nodes that is, we devise a way to protect confidentiality and integrity of data from sensors and queries (modeled as range queries) from the sink.

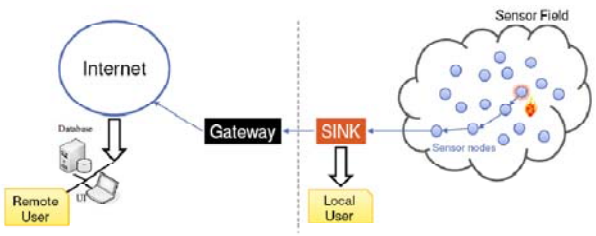

Figure 1: Architecture of wireless sensor network

2. Literature Survey

2.1 Verifiable Privacy-Preserving Range Query in Two-tiered Sensor Networks

By Bo Sheng and Qun Li (IEEE Transaction on Network Security, 2011

- In a sensor network that is not fully trusted and ask the question how we preserve privacy for the collected data and how we verify the data reply from the network.
- Bucketing scheme to mix the data for a range, use message encryption for data integrity, and employ encoding numbers to prevent the storage nodes from dropping data has been used.

3. System Analysis

3. 1 Existing System

The solution to this problem was proposed by Sheng and li which is called as “S&L scheme”. This scheme has two main drawbacks:

- It allows attackers from gaining information collected by sensors and also the queries issued by the sink.
- With increase in the number of dimensions the power consumption and storage space for storage nodes and sensors increases.

Disadvantages

- It allows attackers to obtain a reasonable estimation on both sensor collected data and sink issued queries; and
- The power consumption and storage space for both sensors and storage nodes grow exponentially with the number of dimensions of collected data**.**

3.2 Proposed System

The proposed scheme to preserve privacy and integrity of range queries in sensor networks uses the bucket-partitioning 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. 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. We also propose a solution to adapt SafeQ for event-driven sensor networks, where a sensor submits data to its nearby storage node only when a certain event happens.

Advantages:

- SafeQ provides significantly better security and privacy.
- SafeQ delivers orders of magnitude better performance on both power consumption and storage space for multidimensional data.

4. Implementation

Implementation is the stage of the project when the theoretical design is turned out into a working system. Thus it can be considered to be the most critical stage in achieving a successful new system and in giving the user, confidence that the new system will work and be effective.

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. 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 the storage node sends the result s back to the sink. The sink unifies the query answers which are obtained from multiple storage nodes into a single answer and sends them to the user.

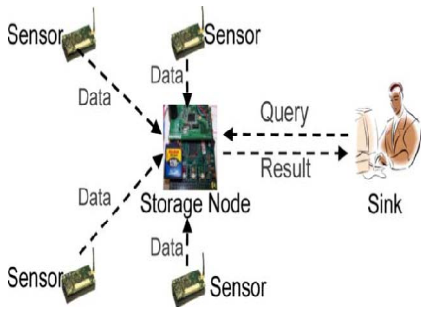

Figure 2: two tiered sensor network

5. Modules

1. Privacy and integrity preserving in WSN's.
2. Privacy for One-dimensional data.
3. Integrity for One-dimensional data.
4. Queries over Multidimensional data.
5. SafeQ Optimization.

5.1 Privacy and Integrity preserving in WSN's

The proposed optimized versions of S&L's integrity preserving scheme aiming to reduce the communication cost between sensors and storage node. The basic idea of their optimization is that each sensor uses a bitmap to represent which bucket have idea and broadcasts its bitmap to the nearby sensors. Each sensor attaches the bitmaps received from others to its own data items and encrypts them together. The sink verifies query result integrity for a sensor by examining the bitmaps.

5.2. Privacy for One-Dimensional Data

To preserve privacy, Sensor encrypts data and sink encrypt queries. The storage node processes encrypted queries over encrypted data.

5.2.1. Prefix Membership Verification

The idea of this to convert the verification of whether a number is in a range to several verifications of whether two numbers are equal.

5.2.2. Submission Protocol:

The submission protocol concerns how a sensor sends its data to a storage node.

5.2.3. Query protocol

The query protocol concerns how the sink sends a range query to a storage node.

5.2.4. Query Processing:

Upon receiving query, the storage node processes this query on the data items received from each nearby sensor at time-slot.

5.3. Integrity for one dimensional Data

To allow the sink to verify the integrity of a query result, the query response from a storage node to the sink consists of two parts**:**

- The query result QR, which includes all the encrypted data items that satisfy the query;
- The verification object VO, which includes information for the sink to verify the integrity of QR.

To achieve this purpose, we propose two schemes based on two different techniques:

- Integrity scheme using merkle hash trees.
- Integrity scheme using neighborhood chains.

5.3.1. Integrity using merkle hash trees

- Each time a sensor sends data items to storage nodes; it constructs a merkle hash tree for the data items.

- Upon receiving a query result QR and its verification object, the sink computes the root value of the merkle hash tree and then verifies the integrity of query result.

5.3.2. Integrity scheme using Neighborhood chains

- First, the sink verifies that query item in QR satisfies the query.
- Second, the sink verifies that the storage node has not excluded any items that satisfy the query.

5.4. Queries over multidimensional data

Sensor collected data and sink issued queries are typically multidimensional as most sensors are equipped with multiple sensing modules such as temperature, humidity, pressure, etc.

5.4.1. Privacy for Multidimensional Data

- The sensor collects the data items with in a timeslot.
- Sensor encrypts data using secret key and for each dimension, sensor applies H function and obtains an encrypted data.
- Sensor sends the encrypted data to nearby storage node.
- When sink wants to perform query on a storage node, the sink applies the G function on each sub query and sends to the storage node.

5.4.2. Integrity for Multidimensional Data

Two integrity preserving schemes for multidimensional data: one uses a Merkle hash tree for each dimension, and other uses a multidimensional neighborhood chain.

a. Integrity Scheme using Merkle hash Trees

For a storage node that is near to sensor, each time it receives a query; it first finds the query result for each range.

- Second, it chooses the query result that contains the smallest number of encrypted data items.
- Third, it computes the merkle hash tree in which the data items are sorted according to the attribute.
- Finally, it sends Query Result and the corresponding verification object to the sink.

b. Integrity Scheme using Neighborhood chains

- The basic idea is that for each of the values in a data items, we find its nearest left neighbor along each dimension and embed this information when we encrypt the item.
- Such neighborhood information is used by the sink for integrity verification.

6. SafeQ optimization:

- This optimization technique is based on Bloom filters to reduce the communication cost between sensors and the storage node.
- A sensor only needs to send the Bloom filter instead of the hashes to a storage node.
- The number of bits needed to represent the Bloom filter is much smaller than that needed to represent the hashes.

7. Algorithm

Algorithm for SHA-1

In cryptography, SHA-1 is a cryptographic hash function . SHA-1 is the most widely used of the existing SHA hash functions, and is employed in several widely used applications and protocols.

SHA-1(160 bit message framework):

- Step 1: Append Padding Bits….

Message is "padded" with a 1 and as many 0's as necessary to bring the message length to 64 bits less than an even multiple of 512.

- Step 2: Append Length....
  64 bits are appended to the end of the padded message. These bits hold the binary format of 64 bits indicating the length of the original message.
- Step 3: Prepare Processing Functions….
  SHA1 requires 80 processing functions defined as:
  f (t; B,C,D) = (B AND C) OR ((NOT B) AND D) (0 <= t <= 19)
  f (t; B,C,D) = B XOR C XOR D (20 <= t <= 39)
  f (t; B,C,D) = (B AND C) OR (B AND D) OR (C AND D) (40 <= t <=59)
  f (t;B,C,D) = B XOR C XOR D (60 <= t <= 79)
- Step 4: Prepare Processing Constants....
  SHA1 requires 80 processing constant words defined as:
  K (t) = 0x5A827999 (0 <= t <= 19)
  K (t) = 0x6ED9EBA1 (20 <= t <= 39)
  K (t) = 0x8F1BBCDC (40 <= t <= 59)
  K (t) = 0xCA62C1D6 (60 <= t <= 79)
- Step 5: Initialize Buffers….
  SHA1 requires 160 bits or 5 buffers of words (32 bits):
  H0 = 0x67452301
  H1 = 0xEFCDAB89
  H2 = 0x98BADCFE
  H3 = 0x10325476
  H4 = 0xC3D2E1F0
- Step 6: Processing Message in 512-bit blocks (L blocks in total message)…. This is the main task of SHA1 algorithm which loops through the padded and appended message in 512-bit blocks.
- Input and predefined functions: M[1, 2, ..., L]: Blocks of the padded and appended message f(0;B,C,D), f(1,B,C,D), ..., f(79,B,C,D): 80 Processing Functions K(0), K(1), ..., K(79): 80 Processing Constant Words. H0, H1, H2, H3, H4, H5: 5 Word buffers with initial values
- Step 6: Pseudo Code….
  For loop on k = 1 to L
  (W (0), W (1),…, W (15)) = M[k] */* Divide M[k] into 16 words */*
  For t = 16 to 79 do:
  W (t) = (W (t-3) XOR W(t-8) XOR W(t-14) XOR W(t-16)) <<< 1
  A = H0, B = H1, C = H2, D = H3, E = H4
  For t = 0 to 79 do:
  TEMP = A<<<5 + f(t;B,C,D) + E + W(t) + K(t) E = D, D = C,
  C = B<<<30, B = A, A = TEMP

End of for loop
H0 = H0 + A, H1 = H1 + B, H2 = H2 + C, H3 = H3 + D, H4 = H4 + E
End of for loop
Output:
H0, H1, H2, H3, H4, H5: Word buffers with final message digest

8. Conclusion

SafeQ prevents a compromised storage node from obtaining a reasonable estimation on the actual values of sensor collected data items and sink issued queries. In terms of efficiency, our results show that SafeQ significantly outperforms prior art for multidimensional data in terms of both power consumption and storage space.

9. Future work

An optimization technique using Bloom filters to significantly reduce the communication cost between sensors and storage nodes. We propose a solution to adapt SafeQ for event-driven sensor networks.

References

[1] F. Chen and A. X. Liu, "SafeQ: Secure and efficient query processing in sensor networks," in Proc. IEEE INFOCOM, 2010, pp. 1–9.
[2] S. Ratnasamy, B. Karp, S. Shenker, D. Estrin, R. Govindan, L. Yin, and F. Yu, "Data-centric storage in sensornets with GHT, a geographic hash table," Mobile Netw. Appl., vol. 8, no. 4, pp. 427–442, 2003.
[3] P. Desnoyers, D. Ganesan, H. Li, and P. Shenoy, "Presto: A predictive storage architecture for sensor networks," in Proc. HotOS, 2005, p. 23.
[4] D. Zeinalipour-Yazti, S. Lin, V. Kalogeraki, D. Gunopulos, and W. A. Najjar, "Microhash: An efficient index structure for flash-based sensor devices," in Proc. FAST, 2005, pp. 31–44.
[5] B. Sheng and Q. Li, "Verifiable privacy-preserving range query in two tiered sensor networks," in Proc. IEEE INFOCOM, 2008, pp. 46–50
[6] J. Shi, R. Zhang, and Y. Zhang, "Secure range queries in tiered sensor networks," in Proc. IEEE INFOCOM, 2009,pp 945–953

Author Profile

R. Swathi received the B.E in Computer Science Engineering in 2011 from BIT Institute of technology, JNTUA University, Anantapur, Andhra Pradesh, India. Currently she is pursuing her M.Tech in Computer Science Engineering in Intell Engineering College, Anantapur, Andhra Pradesh, India.

A. L. Sreenivasulu currently pursuing his PhD in Computer Science in JNTU Anantapur. He got 14 years of teaching experience and guided several projects for both UG & PG students. His areas of interest include Wireless Sensor Networks, Computer Networks and Information Security.