IJREAT International Journal of Research in Engineering & Advanced Technology, Volume 1, Issue 1, March, 2013
ISSN: 2320 - 8791
www.ijreat.org

Collection of Queries in Sensor Networks with Confidentiality and Integrity

Kumar Arun.A[1] and Pandian Arun.R[2]

[1,2] Information Technology,
SAEC Chennai , India

Abstract

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 this paper, we propose SafeQ, a protocol that prevents attackers from gaining information from both sensor collected data and sink issued queries

I. Introduction

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 values. To preserve integrity, we propose two schemes—one using Merkle hash trees and another using a new data structure called neighborhood chains—to generate integrity verification information so that a sink can use this information to verify whether the result of a query contains exactly the data items that satisfy the query.

WIRELESS sensor networks (WSNs) have been widely deployed for various applications, such as environment sensing, building safety monitoring, earthquake prediction, 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. Although important, the privacy- and integrity-preserving range query problem has been under investigated. The prior art solution to this problem was proposed by Sheng and Li in their recent seminal work we call it the "S&L scheme." This scheme has two Main drawbacks:
1) it allows attackers to obtain a reasonable estimation on both sensor collected data and sink issued queries and 2) the power consumption and storage space for both sensors and storage nodes grow exponentially with the number of dimensions of collected data. 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. To preserve integrity, we propose two schemes—one using Merkle hash trees and another using a new data structure called neighborhood chains—to generate integrity verification information such that a sink can use this
Information to verify whether the result of a query contains exactly the data items that satisfy the query. We also propose an optimization technique using Bloom filters to significantly reduce the communication cost between sensors and storage nodes. Furthermore, we 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 and the event may occur infrequently.

SafeQ excels state-of-the-art S&L scheme in two aspects. First, SafeQ provides significantly better security and privacy. While prior art allows a compromised storage node to obtain a reasonable estimation on the value of sensor collected data andsink issued queries, SafeQ makes such estimation very difficult. Second, SafeQ delivers orders of magnitude better performance on both power consumption and storage space for multidimensional data, which are most common in practice as most sensors are equipped with multiple sensing modules such as temperature, humidity, pressure, etc. For power Consumption, for three-dimensional data, SafeQ consumes 184.9 times less power for sensors and 76.8 times less power for storage nodes. For space consumption on storage nodes, for three-dimensional data, SafeQ uses 182.4 times less space. Our experimental results conform with the analysis that the power and space consumption in the S&L scheme grow exponentially with the number of dimensions, whereas those in SafeQ grow linearly with the number of dimensions times the number of data items.

II. RELATED WORK

*A. Privacy and Integrity Preserving in WSNs*Privacy- and integrity-preserving range queries in WSNs Have drawn

people's attention recently Sheng and Li proposed a scheme to preserve the privacy and integrity of range queries in sensor networks 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, 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 S&L scheme has two main drawbacks inherited from the bucket-partitioning technique. First, as pointed out in, the bucket-partitioning technique allows compromised storage nodes to obtain a reasonable estimation on the actual value of both data items and queries. In SafeQ, such estimations are very difficult. 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. In SafeQ, power and space consumption increases linearly with the number of dimensions times the number of data items. Shi *et al.* proposed an optimized version of S&L's integrity preserving scheme aiming to reduce the communication cost between sensors and storage nodes. The basic idea of their optimization is that each sensor uses a bit map to represent which buckets have data and broadcasts its bit map to the nearby Sensors. Each sensor attaches the bit maps received from others to its own data items and encrypts them together. The sink verifies query result integrity for a sensor by examining the bit maps from its nearby sensors. In our experiments, we did not choose The solutions in [11] and [12] for side-by-side comparison for two reasons. First, the techniques used in [11] and [12] are similar to the S&L scheme except the optimization for integrity verification.The way they extend the S&L scheme to handle Multidimensional data is to divide the domain of each dimension into multiple buckets. They inherit the same weakness of allowing compromised storage nodes to estimate the values of data items and queries with the S&L scheme. Second, their optimization technique allows a compromised sensor to easily compromise the integrity verification functionality of the network by sending falsified bit maps to sensors and storage nodes. In contrast, in S&L and our schemes, a compromised sensor cannot jeopardize the querying and verification of data collected by other sensors.

B. Privacy Preserving in Databases

Database privacy has been studied in prior work [13]–[17]. Hacigumus *et al.* first proposed the bucket partitioning idea for querying encrypted data in the database-as-service model (DAS), where sensitive data are outsourced to an untrusted server [13]. Agrawal *et al.* further used the bucket-partitioning idea to investigate range queries on numerical data [15].
Hore *et al.* explored the optimal partitioning of buckets [14]. However, they have the same two drawbacks as we discussed above. Boneh and Waters proposed a public-key system for supporting conjunctive, subset, and range queries on encrypted data [18]. Although theoretically this seems possible, Boneh and Waters's scheme cannot be used to solve our privacy problem because it is too expensive for sensor networks. It would require a sensor to perform encryption for each data submission, where is the number of dimensions and is the domain size (i.e., the number of all possible values) of each dimension. Here, could be large, and each encryption is expensive due to the use of public key cryptography.

C. Integrity Preserving in Databases

Database integrity has also been explored in prior work [19]–[24], independent of the privacy issues. It focuses on verifying the completeness of the result of relational database queries. Merkle hash trees have been used for the authentication of data elements [25], and they were used for verifying the integrity of database queries in [19] and [20]. Pang *et al.* [21] and Narasimha and Tsudik [22] proposed similar schemes for verifying the integrity of relational database query results using signature aggregation and chaining. For each tuple in a database, Pang *et al.* 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 [21]. Narasimha and Tsudik computed the signature by signing the concatenation of the digests of the tuple and its left neighbors along each dimension [22]. 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.

III. MODELS AND PROBLEM STATEMENT

A. System Model

We consider two-tired sensor networks as illustrated in Fig. 1. A two-tired sensor network consists of three types of nodes: *sensors*, *storage nodes*, and a *sink*. 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. 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 We assume that sensors and storage nodes are loosely synchronized with the sink. With loose synchronization, we divide time into fixed duration intervals, and every sensor collects data once per *time interval*. From a starting time that all sensors and the sink agree upon, every time intervals form a *time-slot*. From the same starting time, after a sensor collects data for times, it sends a message that contains a 3-tuple , where is the sensor ID and is the sequence number of the time-slot in which the data items are collected by sensor . We address privacyand integrity-preserving ranges queries for event-driven sensor networks, where a sensor only submits data to a nearby storage node when a certain event happens, in Section IX. We further assume that the queries from the sink are range queries. A range query "finding all the data items collected at time-slot in the range " is denoted as . Note that the queries in most sensor network applications can be easily modeled as range queries. Table I shows the notation used in this paper.

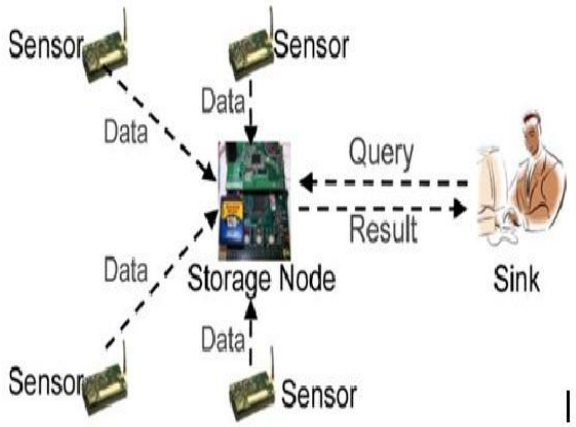

Fig. 1. Architecture of two-tiered sensor networks

B. Threat Model

For a two-tiered sensor network, we assume that the sensors and the sink are trusted, but the storage nodes are not. In a hostile environment, both sensors and storage nodes can be compromised. If a sensor is compromised, the subsequent collected data of the sensor will be known to the attacker, and the compromised sensor may send forged data to its closest storage node. It is extremely difficult to prevent such attacks without the use of tamper-proof hardware. However, the data from one sensor constitute a small fraction of the collected data of the whole sensor network. Therefore, we mainly focus on the scenario Where a storage node is compromised. Compromising a storage node can cause much greater damage to the sensor network than compromising a sensor. After a storage node is compromised, the large quantity of data stored on the node will be known to the attacker, and upon receiving a query from the sink, the compromised storage node may return a falsified result formed by including forged data or excluding legitimate data. Therefore, attackers are more motivated to compromise storage nodes.

C. Problem Statement

The fundamental problem for a two-tired sensor network is the following: *How can we design the storage scheme and the query protocol in a privacy- and integrity-preserving manner?* A satisfactory solution to this problem should meet the following two requirements.

1) *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.

2) *Data integrity:* 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. Besides these two hard requirements, a desirable solution should have low power and space consumption because these wireless devices have limited resources.

IV. PRIVACY FOR ONE-DIMENSIONAL DATA

To preserve privacy, it seems natural to have sensors encrypt data and the sinks encrypt queries. However, the key challenge is how a storage node processes encrypted queries over encrypted data. The idea of our solution for preserving privacy is illustrated in Fig. 2. We assume that each sensor in a network shares a secret key with the sink. For the data items that a sensor collects in time-slot first encrypts the data items using key, the results of which are represented as. Then, applies a "magic" function to the data items and obtains. The message that the sensor sends to its closest storage node includes both the encrypted data and the associative information. When the sink wants to perform query on a storage node, the sink applies another "magic" function on the range and sends to the storage node. The storage node processes the query over encrypted data collected at time-slot using another "magic" function. The three "magic" functions, and satisfy the following three conditions.

1) A data item is in range if and only if is true. This condition allows the storage node to decide whether should be included in the query result.
2) the *verification object* , which includes information for the sink to verify the integrity of . To achieve this

urpose, we propose two schemes based on two different techniques:
Merkle hash trees and neighborhood chains.

Complexity Analysis

Assume that a sensor collects -dimensional data items in a time-slot, each attribute of a data item is a -bit number, and the HMAC result of each numerical zed prefix is a number. The computation cost, communication cost, and storage space of SafeQ are described in Table II. Note that the communication cost denotes the number of bytes sent for each submission or query, and the storage space denotes the number of bytes stored in a storage node for each submission. Furthermore, note that whether sensor nodes report to storage nodes periodically or upon some events has no impact on these costs of one time sending of data items.

Privacy Analysis

In a SafeQ protected two-tiered sensor network, compromising a storage node does not allow the attacker to obtain the actual values of sensor collected data and sink issued queries. The correctness of this claim is based on the fact that the hash functions and encryption algorithms used in SafeQ are secure. In the submission protocol, a storage node only receives encrypted data items and the secure hash values of prefixes converted from the data items. Without knowing the keys used in the encryption and secure hashing, it is computationally infeasible to compute the actual values of sensor collected data and the corresponding prefixes. In the query protocol, a storage node only receives the secure hash values of prefixes converted from a range query. Without knowing the key used in the secure hashing, it is computationally infeasible to compute the actual values of sink issued queries.

Expecting Results

The experimental results from our side-by-side comparison show that SafeQ significantly outperforms the S&L scheme for multidimensional data in terms of power and space consumption. For the two integrity-preserving schemes, the neighborhood- chaining technique is better than Merkle hash tree technique in terms of both power and space consumption. The rationale for us to include the Merkle hash-tree-based scheme is that Merkle hash trees are the typical approach to achieving integrity. For power consumption, SafeQ-NC+ consumes about the same power for sensors and 0.7 times less power for storage nodes; SafeQ-MHT+ consumes about the same power for sensors and 0.3 times less power for storage nodes; SafeQ-NC consumes 1.0 times more power for sensors and 0.7 times less power for storage nodes; and SafeQ-MHT consumes 1.0 times more power for sensors and 0.3 times less power for storage nodes. For space consumption on storage nodes, SafeQ-NC+ and SafeQ-MHT+ consume about the same space, and SafeQ-NC and SafeQ-MHT consume about 1.0 times more space. show that the power and space savings of SafeQ over prior art grow exponentially with the number of dimensions. For power consumption, for three-dimensional data, SafeQ consumes 184.9 times less power for sensors and 76.8 times less power for storage nodes. For space consumption on storage nodes, for three-dimensional data, SafeQ uses 182.4 times less space. Our experimental results conform with the analysis that the power and space consumption in the S&L scheme grow exponentially with the number of dimensions, whereas those in SafeQ grow in early with the number of dimensions times the number of data items.

CONCLUSION

We make three key contributions in this paper. First, we propose SafeQ, a novel and efficient protocol for handling range queries in two-tiered sensor networks in a privacy- and integrity- preserving fashion. SafeQ uses the techniques of prefix membership verification, Merkle hash trees, and neighborhood chaining. In terms of security, SafeQ significantly strengthens the security of two-tiered sensor networks. Unlike prior art, 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. Second, we propose an optimization technique using Bloom filters to significantly reduce the communication cost between sensors and storage nodes. Third, 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, Q. Li, and W. Mao, “Data storage placement in sensor networks,”
in *Proc. ACM MobiHoc*, 2006, pp. 344–355.

[6] B. Sheng, C. C. Tan, Q. Li, and W. Mao, "An approximation algorithm
for data storage placement in sensor networks," in *Proc. WASA*, 2007,
pp. 71–78.
[7] B. Sheng and Q. Li, "Verifiable privacy-preserving range query in twotiered
sensor networks," in *Proc. IEEE INFOCOM*, 2008, pp. 46–50.
[8] Xbow, "Stargate gateway (spb400)," 2011 [Online]. Available: http://
www.xbow.com
[9] W. A. Najjar, A. Banerjee, and A. Mitra, "RISE:More powerful, energy
efficient, gigabyte scale storage high performance sensors," 2005
[Online]. Available: http://www.cs.ucr.edu/~rise
[10] S. Madden, "Intel lab data," 2004 [Online]. Available: http://berkeley.
intel-research.net/labdata
[11] J. Shi, R. Zhang, and Y. Zhang, "Secure range queries in tiered sensor
networks," in *Proc. IEEE INFOCOM*, 2009, pp. 945–953.
[12] R. Zhang, J. Shi, and Y. Zhang, "Secure multidimensional range
queries in sensor networks," in *Proc. ACM MobiHoc*, 2009, pp.
197–206.
[13] 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.
[14] B. Hore, S. Mehrotra, and G. Tsudik, "A privacy-preserving index for
range queries," in *Proc. VLDB*, 2004, pp. 720–731.
[15] R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu, "Order preserving
encryption for numeric data," in *Proc. ACM SIGMOD*, 2004, pp.
563–574.
[16] D. X. Song, D. Wagner, and A. Perrig, "Practical techniques for
searches on encrypted data," in *Proc. IEEE S&P*, 2000, pp. 44–55.
[17] P. Golle, J. Staddon, and B. Waters, "Secure conjunctive keyword
search over encrypted data," in *Proc. ACNS*, 2004, pp. 31–45.
[18] D. Boneh and B. Waters, "Conjunctive, subset, and range queries on
encrypted data," in *Proc. TCC*, 2007, pp. 535–554.

[19] P. Devanbu, M. Gertz, C. Martel, and S. G. Stubblebine, "Authentic
data publication over the internet," *J. Comput. Security*, vol. 11, no. 3,
pp. 291–314, 2003.
[20] H. Pang and K.-L. Tan, "Authenticating query results in edge computing,"
in *Proc. ICDE*, 2004, p. 560.
[21] H. Pang, A. Jain, K. Ramamritham, and K.-L. Tan, "Verifying completeness
of relational query results in data publishing," in *Proc. ACM SIGMOD*, 2005, pp. 407–418.
[22] M. Narasimha and G. Tsudik, "Authentication of outsourced databases
using signature aggregation and chaining," in *Proc. DASFAA*, 2006, pp.
420–436.
[23] W. Cheng, H. Pang, and K.-L. Tan, "Authenticating multi-dimensional
query results in data publishing," in *Proc. DBSec*, 2006, pp. 60–73.
[24] H.Chen, X.Man,W.Hsu, N. Li, and Q.Wang, "Access control friendly
query verification for outsourced data publishing," in *Proc. ESORICS*,
2008, pp. 177–191.
[25] R. Merkle, "Protocols for public key cryptosystems," in *Proc. IEEE
S&P*, 1980, pp. 122–134.
[26] E.-J. Goh, H. Shacham, N. Modadugu, and D. Boneh, "Sirius: Securing
remote untrusted storage," in *Proc. NDSS*, 2003, pp. 131–145.
[27] M. Kallahalla, E. Riedel, R. Swaminathan, Q. Wang, and K. Fu, "Plutus: Scalable secure file sharing on untrusted storage," in *Proc. FAST*, 2003, pp. 29–42.
[28] J. Cheng, H. Yang, S. H.Wong, and S. Lu, "Design and implementation
of cross-domain cooperative firewall," in *Proc. IEEE ICNP*, 2007, pp.
284–293.

[28] J. Cheng, H. Yang, S. H.Wong, and S. Lu, "Design and implementation
of cross-domain cooperative firewall," in *Proc. IEEE ICNP*, 2007, pp.
284–293.

[29] A. X. Liu and F. Chen, "Collaborative enforcement of firewall policies
in virtual private networks," in *Proc. ACM PODC*, 2008, pp. 95–104.
[30] P. Gupta and N. McKeown, "Algorithms for packet classification,"
IEEE Netw., vol. 15, no. 2, pp. 24–32, Mar.–Apr. 2001.
[31] Y.-K. Chang, "Fast binary and multiway prefix searches for packet forwarding,"
Comput. Netw., vol. 51, no. 3, pp. 588–605, 2007.
[32] H. Krawczyk,M. Bellare, and R. Canetti, "HMAC: Keyed-hashing for
message authentication," RFC 2104, 1997.
[33] R. Rivest, "The md5 message-digest algorithm," RFC 1321, 1992.
[34] D. Eastlake and P. Jones, "Us secure hash algorithm 1 (sha1)," RFC
3174, 2001.
[35] B. Bloom, "Space/time trade-offs in hash coding with allowable errors,"
Commun. ACM vol. 13, no. 7, pp. 422–426, 1970.
[36] P. Levis, "Simulating TinyOS networks," 2003 [Online]. Available:
http://www.cs.berkeley.edu/~pal/research/tossim.html