Effective Mechanism for Secure Query Processing Using an Efficient Protocol

<< N. Vijaya Lakshmi >> [1], <<G.Lavanya >> [*2]
Dept. of Computer Science and Engineering

[#]<< Anurag Group of Institutions >>
Venkatapur (V), Ghatkesar (M), R.R. Dist., Hyderabad, INDIA.

Abstract— ***The architecture associated with two-tiered sensor sites, where storage space nodes serve as an intermediate rate between sensors and also a sink pertaining to storing info and finalizing queries, has been widely adopted due to benefits associated with power along with storage preserving for sensors plus the efficiency associated with query finalizing. However, the importance of storage space nodes in addition makes them popular with attackers. In this project, many of us propose Offered system, the protocol that prevents attackers from attaining information through both sensor accumulated data along with sink given queries. Proposed process also makes it possible for a drain to find compromised storage space nodes once they misbehave. In order to preserve level of privacy, Proposed system runs on the novel method to encode equally data along with queries such that a storage space node can certainly correctly practice encoded requests over encoded info without realizing their ideals. To sustain integrity, we propose two schemes—one applying Merkle hash woods and another utilizing a new info structure called neighborhood chains—to create integrity proof information in order that a sink can use this data to confirm whether the effect of a issue contains the data goods that satisfy your query..***

Keywords— **Integrity, Privacy, Sensor Networks, Virtual Private Network (VPN).**

I. INTRODUCTION

Wi-fi sensor networks are widely started for a variety of applications, for example environment realizing, building protection monitoring, earthquake conjecture, etc. In this project, we consider a two-tiered sensor community architecture by which storage nodes accumulate data through nearby sensors and reply queries from your sink in the network. The storage devices nodes serve for intermediate tier between your sensors and the sink regarding storing information and control queries. Storage nodes take three primary benefits in order to sensor cpa networks. First, sensors conserve power by sending just about all collected data for their closest storage devices node as opposed to sending the crooks to the drain through lengthy routes. Minute, sensors is usually memory-limited mainly because data are mainly saved on storage devices nodes. Next, query control becomes extremely effective because the sink merely communicates using storage nodes regarding queries. The supplement of storage devices nodes inside sensor networks was initially introduced and has been broadly adopted. Numerous products regarding storage nodes, for example Star Gate in addition to RISE, are commercially readily available. However, the inclusion regarding storage nodes furthermore brings major security challenges. As storage devices nodes retail store data received from sensors and serve for important part for giving answers to queries, they may be more vulnerable to be sacrificed, especially in the hostile setting. A sacrificed storage node imposes major threats to your sensor community. First, the assailant may obtain sensitive data that's been, or will probably be, stored from the storage node. Minute, the sacrificed storage node may perhaps return forged data for a query. Next, this storage devices node may well not include just about all data goods that satisfy the query.

Thus, we would like to design a new protocol of which prevents attackers from getting information through both sensor gathered data in addition to sink supplied queries, which typically is usually modelled as range questions, and will allow the drain to diagnose compromised storage devices nodes after they misbehave. With regard to privacy, compromising a new storage node must not allow the attacker to uncover the sensitive information that's been, and will probably be, stored from the node, plus the queries which the storage node features received, and definitely will receive. Realize that we take care of the queries from your sink as confidential mainly because such questions may drip critical info on query issuers' likes and dislikes, which have to be protected specially in military applications. With regard to integrity, the sink must detect whether a query originate from a storage devices node includes forged information items or doesn't include all the data that satisfy the query. You'll find two critical challenges inside solving the privacy in addition to integrity-preserving array query problem. First, a storage devices node must correctly process encoded questions over encoded information without figuring out their genuine values. Minute, a sink must verify that the consequence of a dilemma contains all the data goods that satisfy the query and doesn't contain virtually any forged information. Although crucial, the privacy- in addition to integrity-preserving array query problem has been under perused.

The prior art solution to this problem was proposed by Sheng and Li intheir recent seminal work. We call it the "S&L scheme."This scheme has two main drawbacks: 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. In this project, we propose Proposed system, a novel privacy- and integrity-preserving range query protocol for two-tiered sensor networks. The ideas of proposed system are fundamentally different from the S&L scheme. To preserve privacy, proposed system 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.

II. Previous Work

Virtual Private Network (VPN) is a widely deployed technology that allows roaming users to securely use a remote computer on the public Internet as if that computer is residing on their organization's network, which henceforth allows roaming users to access some resources that are only accessible from their organization's network. VPN works in the following manner. Suppose IBM sends a field representative to one of its customers, say Michigan State University (MSU). Assume that MSU's IP addresses range 1.1.0.0 to 1.1.255.255 and IBM's IP addresses range 2.2.0.0 to 2.2.255.255. To access resources (say a confidential customer database server with IP address 2.2.0.2) that are only accessible within IBM's network, the IBM representative uses an MSU computer (or his laptop) with an MSU IP address (say 1.1.0.10) to establish a secure VPN tunnel to the VPN server (with IP address 2.2.0.1) in IBM's network. Upon establishing the VPN tunnel, the IBM representative's computer is temporarily assigned a virtual IBM IP address (say 2.2.0.25).

Using the VPN tunnel, the IBM representative can access any computer on the Internet as if his computer is residing on IBM's network with IP address 2.2.0.25. The payload of each packet inside the VPN tunnel is another packet (to or from the newly assigned IBM IP address 2.2.0.25), which is typically encrypted. Figure 1 illustrates an example packet that traverses from the IBM representative's computer on MSU's network to the customer database server in IBM's network.

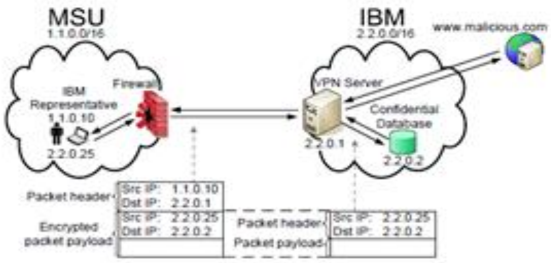


Figure 1: Typical Example

While the VPN tunnel is very useful for the IBM representative, it imposes security threats to MSU’s network because MSU’s firewall does not know what traffic is flowing inside the VPN tunnel. For example, if MSU’s firewall blocks access to a remote site (say www.malicious.com) or disallows machines to run peer-to-peer applications due to copyright concerns, MSU’s firewall cannot enforce its policies on the IBM representative’s computer although that computer is physically on MSU’s network. Basically, the VPN tunnel opens a hole to MSU’s firewall that may allow unwanted traffic to flow inside or outside. Having such a hole is remarkably dangerous because viruses or worms could flood in through it to the IBM representative’s computer first and then further spread to other computers on MSU’s network.

This problem is conceivably difficult to solve for many reasons. First, MSU cannot simply block VPN connections because otherwise the IBM representative may fail to perform his duties. Second, MSU cannot share its firewall policy with IBM. It is common in practice that firewall policies are kept confidential due to security and privacy concerns. Knowing the firewall policy of a network could enable attackers to easily spot the security holes in the policy and launch corresponding attacks. A firewall policy also reveals the IP addresses of important servers, which are usually kept confidential to reduce the chance of being attacked.

Furthermore, from a firewall policy one may be able to derive the business relationship of the organization with their partners. Third, IBM cannot share the traffic in its VPN tunnel with MSU due to security and privacy concerns. For example, IBM may want to keep the IP address of its customer database server confidential to reduce the likelihood of being attacked. One main purpose of VPN is to achieve such confidentiality. [1]

The amount of data stored in databases is rapidly increasing in today's world. A lot of such data is published over the Internet or large-scale intranets. Given the large sizes of databases and the high frequency of queries, it is often desirable for data owners to outsource data publishing to internal or external publishers. In such a model, the data center of an organization gives datasets to internal publishers (for publishing within the organization's intranet), or external third- party publishers. Overloading the task of data publishing from data owners to dedicated data publishers offers the following advantages: the publishers may have higher bandwidths; the publishers may be geographically closer to the clients and have lower latencies; having multiple publishers helps to avoid the data owner being a single point of failure; overall data management cost can be significantly reduced, by leveraging hardware and software solutions from dedicated data publishing service providers.

In many settings the trustworthiness of the data publishers cannot be guaranteed the security of the publishers' servers is not under the control of the data owners. Historical computer security incidents have shown that securing large online systems is a difficult task. The threat of insider attacks from within a data publishing service provider cannot be overlooked either. Therefore it is critical for a client to be able to verify the correctness of query results.

There are three aspects of correctness: authenticity, completeness and freshness. A query result is authentic if all the records in the result are from the dataset provided by the data owner. A query result is complete if the publisher returns all data records that satisfy the query criteria. A query result is fresh if the query result reflects the current state of the owner's database.

Most of these solutions use techniques from public-key cryptography. The data owner has a pair of public/private keys. Verification metadata is generated over the dataset using the private key by the owner, and the metadata is provided to the publishers with the dataset. When a client queries from a data publisher, the publisher returns the query result together with a proof called a Verification Object (VO), which is constructed based on such metadata. The client can then verify the correctness of the query result using the corresponding VO, with the data owner's public key.

Due to increasing privacy concerns for today's information management, preserving data privacy has become a critical requirement. The data owner and the publishers need to enforce access control policies so that each client can only access the information within her own accessible area. On the other hand, clients should be able to verify the correctness (namely authenticity, completeness and freshness) of the query results, even if the publishers could be malicious or be compromised. The data privacy should be preserved at least when the publishers are benign. When the publishers are compromised, the bottom line is that the publishers cannot cheat the clients by giving them bogus query results. Though in that case, data privacy might be violated.[2]

Achieving completeness of query results while preserving privacy is challenging. To achieve completeness, one needs to show that there exists no other record in the query region other than the ones that are returned in the query result. Most existing approaches leak information that is outside the query region and the client's accessible area, violating the privacy preserving requirement. For instance, back to the above example. Most existing solutions rely on proving the adjacency of data points. The data owner signs the data pair (20; 30) to indicate that there are no data points between 20 and 30 in the dataset. The publisher can return the signed data pair as part of the proof of completeness. However, in doing so, the data point 30, which is outside the client's access control area, is revealed to the client and thus data privacy is violated. Besides data privacy, we also considers policy privacy, where the client should not know the boundaries of other users' access control areas. In Pang et al. proposed a scheme which allows correctness verification while preserving the privacy of data records. However, the solution applies only to one dimensional range queries, which is a significant limitation given the multidimensional nature of the relational databases today.

Query-answering with third party publishing has been studied in the computer security and cryptography community under the name authenticated dictionary. Schemes using Merkle Tree and skip lists have been proposed; however these approaches assume that the data is public and do not consider the access control requirement. In a scheme based on Merkle Tree is pro- posed to guarantee the authenticity and completeness. In this approach is generalized and applied to other authenticated data structures. Data structures based on space and data partitioning are introduced for verifying multidimensional query results. In these approaches, data privacy is not preserved. A scheme to verify the integrity of the query results in edge computing is proposed. The scheme does not check the completeness of query results. The security model, where semi-trusted service providers answer user queries on behalf of the data owner, is similar to ours.

In order to preserve the data privacy, proposed another scheme to solve the problem. This approach handles one-dimensional case well, but it cannot be applied to two or higher dimensional cases. In the overheads and performances of different approaches to guarantee the authenticity and completeness are compared.

Several approaches are proposed for enforcing access control policies in out- sourced XML documents. The XML documents, which can be viewed as trees, present different structures from relational databases. The data structure and algorithms derived in this paper uses the divide-and- conquer strategy developed in data structures and algorithms for range searching by the computational geometry community. Our approach is unique in that our approach addresses the specific requirements of outsourced data publishing, especially the need to efficiently prove the search results. Also our solution incurs low communication and storage overhead when updating the data items and/or the access policies of clients. [2]

In data publishing, a data owner delegates the role of satisfying user queries to a third-party publisher. This model is applicable to a wide range of computing platforms, including database caching, content delivery network, edge computing, P2P databases, etc. The data publishing model offers a number of advantages over conventional client-server architecture where the owner also undertakes the processing of user queries. By pushing application logic and data processing from the owner's data center out to multiple publisher servers situated near user clusters, network latency can be reduced. Adding publisher servers is also likely to be a cheaper way to achieve scalability than fortifying the owner's data center and provisioning more network bandwidth for every user. Finally, the data publishing model removes the single point of failure in the owner's data center, hence reducing the database's susceptibility to denial of service attacks and improving service availability.

However, since the publishers are outside of the administrative domain of the data owner, and in fact may reside on poorly secured platforms, the query results that they generate cannot be accepted at face value, especially where they are used as basis for critical decisions. Instead, there must be provisions for the user to check the "correctness" of a query answer.

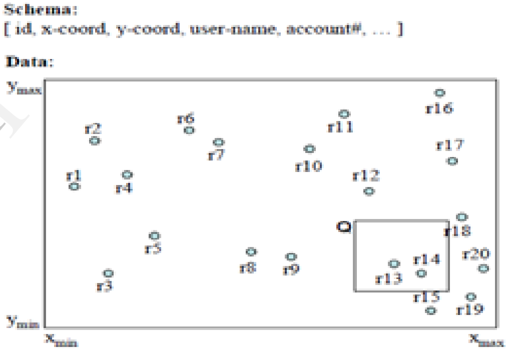


Figure 2: Sample Schema Example

Consider a dataset containing 20 data points in two-dimensional space as shown in Figure above. The figure also includes a window query **Q**, for which *{r*13, *r*14*}* is the correct result. A rogue publisher may return a wrong result *{r*13, *r*14, *r*100*}*, which includes a spurious point *r*100, or *{r*13, *r*14*}* in which some attribute values of *r*13 have been tampered with. To detect such incorrect values, the user should be able to verify the *authenticity* of query result. A different threat is that the publisher may omit some result points, for example by returning only *{r*13*}* for query **Q**. This threat relates to the *completeness* of query result. Most of the existing works provide for checking the authenticity and completeness of query results on *one-dimensional* datasets. The exception is Devanbu's scheme which handles multiple key attributes by essentially concatenating them in some preferred order key1|key2|..|key*d*. However, the scheme is expected to be very inefficient for symmetric queries, such as window and nearest neighbor queries, that are typical in multi-dimensional context.

System and Threat Models

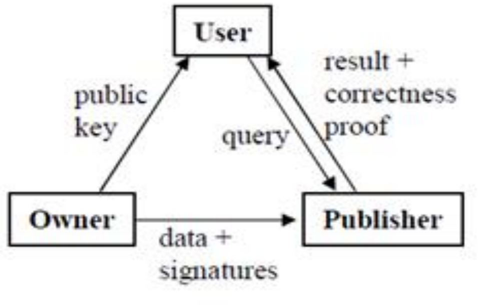


Figure 3: Data Publishing Model

Figure above depicts the data publishing model, which supports three distinct roles: The data owner maintains a master database, and distributes it with one or more associated signatures that prove the authenticity of the database. Any data that has a matching signature is accepted by the user to be trustworthy. The publisher hosts the database, and executes queries on behalf of the owner. There could be several publisher servers that are situated at the edge of the network, near the user applications. The publisher is not required to be trusted, so the query results that it generates must be accompanied by some "correctness proof", derived from the database and signatures issued by the owner. The user issues queries to the publisher explicitly, or else gets redirected to the publisher, e.g. by the owner or a directory service. To verify the signatures in the query results, the user obtains the public key of the owner through an authenticated channel, such as a public key certificate issued by a certificate authority.

Our primary concern addressed in this paper is the threat that a dishonest publisher may return incorrect query results to the users, whether intentionally or under the influence of an adversary. An adversary who is cognizant of the data organization in the publisher server may make logical alterations to the data, thus inducing incorrect query results. Even if the data organization is hidden, for example through data encryption or steganographic schemes, the adversary may still sabotage the database by overwriting physical pages within the storage volume. In addition, a compromised publisher server could be made to return incomplete query results by withholding data intentionally. Therefore mechanisms for users to verify the completeness as well as authenticity of their query results are essential for the data publishing model. While there are several other security considerations in the data publishing model such as privacy, user authentication and access control, these have been studied extensively, and are orthogonal to our work here.

Cryptographic Primitives

Our proposed solution and many of the related work are based on the following cryptographic primitives:

One-way hash function: A one-way hash function, denoted as $h(.)$, is a hash function that works in one direction: it is easy to compute a fixed-length digest $h(m)$ from a variable-length pre-image m; however, it is hard to find a pre-image that hashes to a given hash value. Examples include

MD5 and SHA. We will use the terms hash, hash value and digest interchangeably.

Digital signature: A digital signature algorithm is a cryptographic tool for authenticating the integrity and origin of a signed message. In the algorithm, the signer uses a private key to generate digital signatures on messages, while a corresponding public key is used by anyone to verify the signatures. RSA and DSA are two commonly-used signature algorithms.

Signature aggregation: As introduced, this is a multi-signer scheme that aggregates signatures generated by distinct signers on different messages into one signature. Signing a message *m* involves computing the message hash $h(m)$ and then the signature on the hash value. To aggregate *t* signatures, one simply multiplies the individual signatures, so the aggregated signature has the same size as each individual signature. Verification of an aggregated signature involves computing the product of all message hashes and then matching with the aggregated signature.

Signature chain: In a signature chain scheme is proposed that enables clients to verify the completeness of answers of range queries. A very nice property of the scheme is that only result values are returned, thus ensuring that there is no violation of access control. The scheme is based on two concepts: The signature of a record is derived from its own digest as well as its left and right neighbors'. In this way, an attempt to drop any value from the answer of a range query will be detected since it would no longer be possible to derive the correct signature for the record that depends on the dropped value. For the boundaries of the answer, a collaborative scheme that involves both the publisher and the client is proposed the publisher performs partial computation based on but not revealing the two records bounding the answer and the query range, while the client completes the computation based on the two end points of the query range.

Signature Chain in Multi-Dimensional Space

The goal of our work is to devise a solution for checking the correctness of query answers on multi-dimensional datasets. The design objectives include:

Completeness: The user can verify that all the data points that satisfy a window query are included in the answer.

Authenticity: The user can check that all the values in a query answer originated from the data owner. They have not been tampered with, nor have spurious data points been introduced.

Precision: Proving the correctness of a query answer entails minimal disclosure of data points that lie beyond the query window. We define precision as the ratio of the number of data points within the query window, to the number of data points returned to the user.

Security: It is computationally infeasible for the publisher to cheat by generating a valid proof for an incorrect query answer.

Efficiency: The procedure for the publisher to generate the proof for a query answer has polynomial complexity. Likewise the procedure for the user to check the proof has polynomial complexity.

Without loss of generality, we assume that the data in the multi-dimensional space are split into partitions this can be done using a spatial data structure. To ensure that the answer for a window query is complete, two issues must be addressed. First, we need to prove that the answer covers all the partitions that overlap the query window. We refer to these partitions as candidate partitions. Second, we need to prove that all qualifying values within each candidate partition are returned. [3]

III. Proposed System

A. *Membership Verification*

In this module taking that approach is to convert this verification of whether several is within a range to several verifications of whether two numbers usually are equal. A prefix having k top 0's and also 1's is known as a k-prefix. As an example, 1*** can be a 1-prefix, and it denotes the range [1000, 1111]. If the value by matches a k-prefix, the primary k-bits of x along with the k-prefix include the same. To examine whether several a was in a array [d1, d2], we very first convert the range to the absolute minimum set of prefixes, denoted by means of S([d1, d2]), so that the union in the prefixes is adequate to [d1, d2]. Today we calculate the prefix spouse and children for amount a. To be able to verify whether two volumes are identical, we turn each prefix with a corresponding one of a kind number by using a prefix numericalization function..

B. *Submission Protocol*

The particular submission project concerns how a sensor transmits its data with a storage node. Let d1, d2, d3…. Dn be data items that sensor si collects at a time-slot. Sort your data items in an ascending obtain. Convert your ranges thus to their corresponding prefix counsel, Numericalize almost all prefixes. Compute your keyed Hash Message Authentication Signal (HMAC) of each numericalized prefix employing key, which is known to all sensors and also the sink. Encrypt each data piece with crucial. Sensor transmits the encrypted data along with to their closest storage node..

$$\mathrm{HMAC}_g(\mathcal{N}(\mathcal{F}(a))) \cap \mathrm{HMAC}_g(\mathcal{N}(\mathcal{S}([d_{n_1-1}, d_{n_1}]))) \neq \emptyset$$

$$\mathrm{HMAC}_g(\mathcal{N}(\mathcal{F}(b))) \cap \mathrm{HMAC}_g(\mathcal{N}(\mathcal{S}([d_{n_2-1}, d_{n_2}]))) \neq \emptyset.$$

C. *Query Protocol*

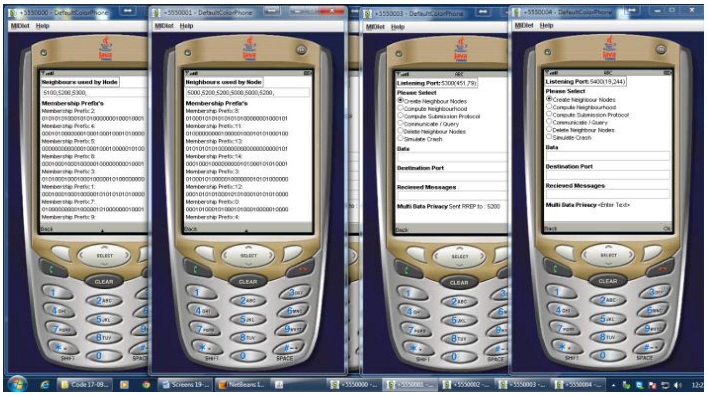
This query method concerns the way the sink sends a selection query to a storage node. When the sink would like to perform query over a storage node, it performs the subsequent steps. Compute prefix people and Numericalize many prefixes, Apply HMAC in order to each numericalized prefix and Send query towards storage node. With the onewayness and collision resistance properties on the HMAC operate, the storage space node are unable to compute any andb in the query it receives..

D. *Multidimensional Data Privacy*

In this particular module we all extend our own privacy-preserving techniques for one-dimensional data to multidimensional data. Let D1, D2, D3 … Dn denote this z-dimensional data items which a sensor gathers at time-slot. First, encrypts these data which consists of secret crucial, second, per dimension does apply the "magic" perform. At very last, sends this encrypted data what to a close by storage node. If the sink wants to perform query on the storage node, the kitchen sink applies this "magic" perform on each sub query along with sends to the storage node. The storage space node next applies this "magic" function to discover the query result per sub query.

IV. Results

The concept of this paper is implemented and different results are shown below, The proposed paper is implemented in Java technology on a Pentium-IV PC with minimum 20 GB hard-disk and 1GB RAM. The propose paper's concepts shows efficient results and has been efficiently tested on different Datasets.



+5550000 - DefaultColorPhone

Neighbours used by Node
5100,5200,5300,
Membership Prefix's
Membership Prefix:2:
01010101000101010000000100010001
Membership Prefix:4:
00010100000001000100010001010000
Membership Prefix:5:
00000000000001000100010000010100
Membership Prefix:8:
00010001000100000100000000010001
Membership Prefix:3:
01010001000100010000010100000000
Membership Prefix:1:
00010001000100000101010101010000
Membership Prefix:7:
01000000000100000101000000010001
Membership Prefix:9:

+5550001 - DefaultColorPhone

Neighbours used by Node
5000,5200,5200,5000,5000,5200,
Membership Prefix's
Membership Prefix:8:
01000101010101010100000001000101
Membership Prefix:11:
01000000000100000100010101000100
Membership Prefix:13:
01010101010000000000000000000101
Membership Prefix:14:
00010001000000000101000101010001
Membership Prefix:3:
01000101000001000000010101000000
Membership Prefix:12:
00010101010001010100010101010000
Membership Prefix:0:
00010100010100010100010000010000
Membership Prefix:4:

+5550003 - DefaultColorPhone

Listening Port: 5300(451,79)
Please Select
● Create Neighbour Nodes
○ Compute Neighbourhood
○ Compute Submission Protocol
○ Communicate / Query
○ Delete Neighbour Nodes
○ Simulate Crash
Data
Destination Port
Recieved Messages
Multi Data Privacy Sent RREP to : 5200

+5550004 - DefaultColorPhone

Listening Port: 5400(19,244)
Please Select
● Create Neighbour Nodes
○ Compute Neighbourhood
○ Compute Submission Protocol
○ Communicate / Query
○ Delete Neighbour Nodes
○ Simulate Crash
Data
Destination Port
Recieved Messages
Multi Data Privacy <Enter Text>

Fig. 4 Proposed system performing membership prefic.

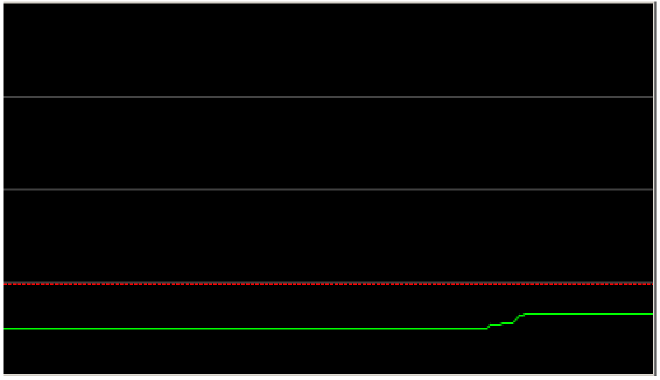
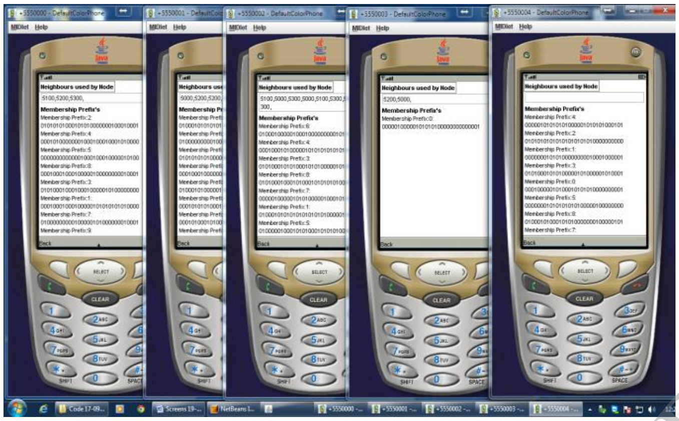


Fig. 7 Time taken by Submission Protocol



+5550000 - DefaultColorPhone

Neighbours used by Node
5100,5200,5300,
Membership Prefix's
Membership Prefix:2:
01010101000101010000000100010001
Membership Prefix:4:
00010100000001000100010001010000
Membership Prefix:5:
00000000000001000100010000010100
Membership Prefix:8:
00010001000100000100000000010001
Membership Prefix:3:
01010001000100010000010100000000
Membership Prefix:1:
00010001000100000101010101010000
Membership Prefix:7:
01000000000100000101000000010001
Membership Prefix:9:

+5550002 - DefaultColorPhone

Neighbours used by Node
5100,5000,5300,5000,5100,5300,5
300,
Membership Prefix's
Membership Prefix:6:
Membership Prefix:4:
Membership Prefix:3:
Membership Prefix:8:
Membership Prefix:7:
Membership Prefix:1:
Membership Prefix:5:

+5550003 - DefaultColorPhone

Neighbours used by Node
5200,5000,
Membership Prefix's
Membership Prefix:0:
00000100000101010100000000000001

+5550004 - DefaultColorPhone

Neighbours used by Node
Membership Prefix's
Membership Prefix:4:
00000101010101000001010101000101
Membership Prefix:2:
01010101010101010101010000000000
Membership Prefix:1:
00000001010100000000010001000001
Membership Prefix:3:
01010001010100000101000001010001
Membership Prefix:0:
00010000010100010101010000000001
Membership Prefix:5:
00000001010101010100000100000000
Membership Prefix:8:
01000101000101010000000100000101
Membership Prefix:7:

Fig. 5 Proposed system displaying membership prefix for all nodes

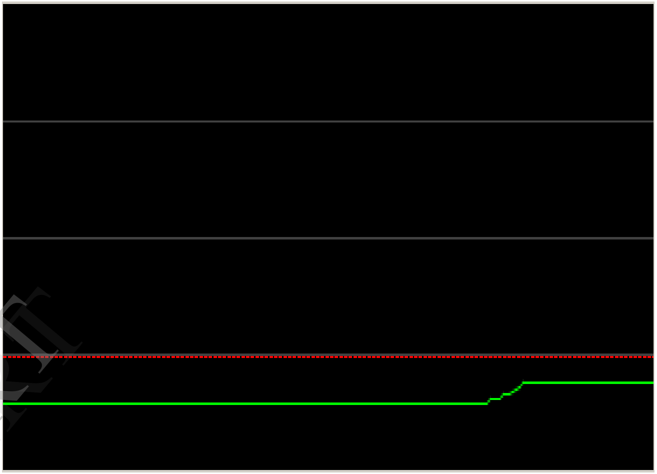


Fig. 8 Time taken by Data Query

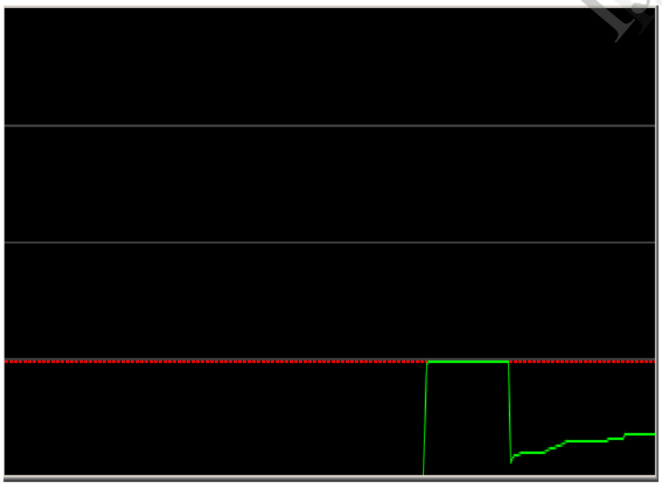


Fig. 6 Time taken by Node to initialize.

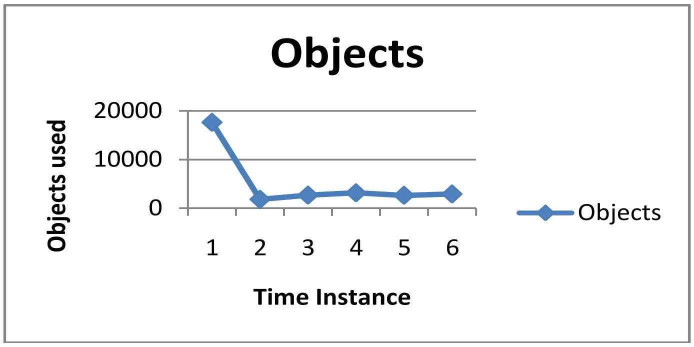


Fig. 9 Time taken by Node to initialize by objects.

Time Instance	Objects
1	17632
2	1800
3	2659
4	3137
5	2629
6	2892

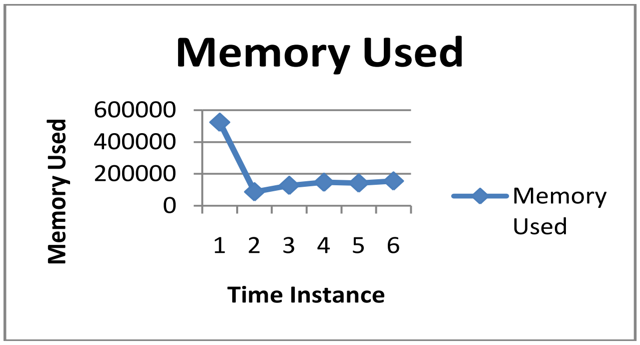


Fig. 10 Time taken by Node to compute Memory Used

Time Instance	Memory Used
1	523752
2	86660
3	127072
4	146904
5	141544
6	154596

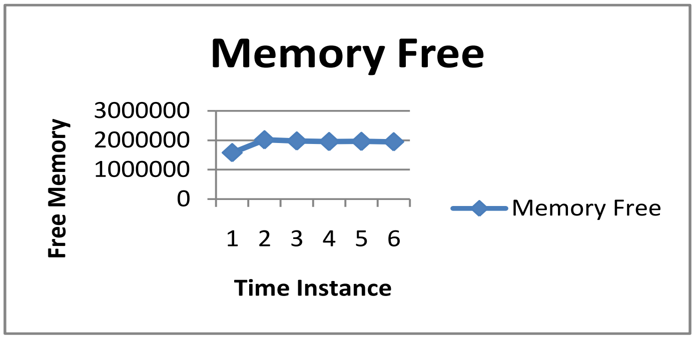


Fig. 11 Time taken by Node Communicating and Free Memory

Time Instance	Memory Free
1	1573400
2	2010492
3	1970080
4	1950248
5	1955608
6	1942556

V. Conclusions

We all make three key contributions in this particular project. 1st, we offer system, a story and efficient protocol intended for handling selection queries throughout two-tiered sensor networks inside a privacy as well as integrity protecting fashion. Planned system makes use of the strategies of prefix account verification, Merkle hash timber, and area chaining. Regarding security, recommended system considerably strengthens the security of two-tiered sensor sites. Unlike before art, Proposed system prevents the compromised hard drive node from obtaining a reasonable estimation on the actual ideals of sensor collected data items and torpedo issued concerns. In phrases of efficiency, our benefits show that proposed system significantly outperforms before art intended for multidimensional data in terms of both electric power consumption and hard drive. Second, we offer an marketing technique making use of Bloom filtration to significantly reduce the communication price between sensors and hard drive nodes. Next, we propose a fix to adapt Proposed system for event-driven sensor sites..

References

[1] A. X. Liu and F. Chen, “Collaborative enforcement of firewall policies in virtual private networks,” in *Proc. ACM PODC*, 2008, pp. 95–104.

[2] 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.

[3] W. Cheng, H. Pang, and K.-L.Tan, "Authenticating multi-dimensional query results in data publishing," in *Proc. DBSec*, 2006, pp. 60–73.

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

[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] 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, Q. Li, and W. Mao, "Data storage placement in sensor networks,"in *Proc. ACM MobiHoc*, 2006, pp. 344–355.

[8] Y.-K. Chang, "Fast binary and multiway prefix searches for packet forwarding,"*Comput.Netw.*, vol. 51, no. 3, pp. 588–605, 2007.

[9] F. Chen and A. X. Liu, "SafeQ: Secure and efficient query processing in sensor networks," in *Proc. IEEE INFOCOM*, 2010, pp. 1–9.

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

[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 rangequeries in sensor networks," in *Proc. ACM MobiHoc*, 2009, pp.197–206.

Mrs N.Vijaya lakshmi, The author, Completed BTech CSIT from Sri Vidya Nikethan Engineering College,A.Rangampet ,Chittoor(Dt) in 2003.

Worked as an Assistant Professor , CSE Dept. in S.K.D Engineering College, Gooty from 2003-2006.

Pursuing MTech CSE in Anurag group of Institutions, Venkatapur (V), Ghatkesar(M),R.R.Dist, Hyderabad from 2011-2013.

Has done research in providing security to Wireless Sensor Networks.

Ms. G. Lavanya Working as an Assistant Professor in Anurag group of Institutions, guiding many of the BTech and MTech students to complete their Research successfully.