Security Aspects of Wireless Heterogeneous Databases - Protocol, Performance, and Energy Analysis[1]

Harshal Haridas, Ali R. Hurson, and Yu Jiao
Department of Computer Science and Engineering
University Park, PA 16802
{haridas, hurson, yjiao}@cse.psu.edu

Abstract

Users have been demanding information "anytime, anywhere". The notion of accessing diverse and autonomous information repositories with different APIs is not accepted. This has necessitated the logical integration of diverse information from different sources with common APIs. Current multi-database researchers have addressed effectively issues such as local autonomy, heterogeneity, transaction management, concurrency control, transparency, and query resolution in a "sometimes, somewhere" environment.

The concept of mobility, however, introduces additional complexities and restrictions to multi-database designers. A user accessing data through a remote connection with a portable device has high probability to face issues like reduced capacity network connections, frequent disconnections, and processing and resource restrictions.

This work extends the scope of our previous effort in the design of an authorization model for a multi-database system. The Authorized Summary Schema Model is used as the underlying paradigm and is extended to provide a secure wireless environment. The proposed system is simulated and the impact of a wireless medium on performance is presented, evaluated, and analyzed.

1. Introduction and Motivation:

The user's need to access diverse and autonomous information repositories through different APIs (Application Program Interfaces) is not accepted since the user has to be retrained to "*learn a new API and its usage on accessing different information sources*".

Research in global information sharing process is designed to offer timely and reliable access to remote data in wired and wireless medium. However, this alone is not sufficient to ensure happy users. Users also require Confidentiality, Integrity, and Availability of this remote data, and Authenticity of accessing the data. Within the scope of global information sharing process as witnessed by the literature, security aspect has received less attention [1, 2, 11, and 12]. Trust is a major factor that needs to be established in creation of a secure environment allowing remote data access by users. Violating trust would result in havoc in the system.

Providing security in distributed systems is however a difficult task. The wireless medium further imposes lower bandwidth, lower computing resources, frequent disconnection, higher error rate, and non-secure links in the face of autonomy and heterogeneity [3]. Site autonomy also includes enforcing local security. Whether security is enforced globally and/or locally a secure global information sharing system must address issues as such; access control, authorizations, and counter-measures for inferential security and accountability. Finally, authorized users should be given access to information based on their role or access rights in the global system.

Within the scope of a global information sharing environment, this paper studies security in a multi-database environment enhanced by wireless communication. We study an authorization model in a sometimes, somewhere environment (multi-database system) and extend it to an anytime, anywhere environment (wireless environment with mobile clients and mobile databases). SSM enhanced with an authorization model [4] is used as the underlying paradigm as a multi-database solution. The remainder of the paper is organized as follows. Section 2 gives background of SSM and authorized SSM. Section 3 identifies basic requirements for a secure wireless transmission and then presents an integrated solution to meet these requirements. Section 4 presents our extended system in detail. The proposed model is simulated and its behavior based on several performance metrics — response time, energy consumption and network traffic — are presented and analyzed in Section 5. Finally, Section 6 concludes our work.

2. An authorization Model for a Multi-database System

2.1. Summary Schema Model

Summary Schemas Model (SSM) is an adjunct to multi-database language systems and is proposed to resolve name differences among similar data in multi-database systems [3]. The word relationships (synonyms, hypernyms, or hyponyms) are used to build a hierarchical meta-data of local access terms, exported from underlying local databases. Consequently, the SSM meta-data hierarchy captures the data semantic globally shared. This allows the global user to submit *imprecise* queries at any

[1] This work in part has been supported by the Office of Naval Research under contratN00014-02-1-0282.

site. The SSM intelligently maps these imprecise query terms with precise access terms found at local databases.

The hierarchical structure of the SSM consists of local database schemas as leaf nodes and summary schemas as internal nodes (Figure 1). A schema at each node is a list of access terms. Mapping access terms of the children nodes to their corresponding hypernyms creates a summary schema node. Summary schemas are hence smaller and more abstract than the union of the lower schemas. Semantic similarity between any two access terms is measured in terms of *Semantic Distance Metric* (SDM) computed using the number of hypernyms-hyponyms and synonyms linked between them [3].

In response to a global query, the search is continued and conducted in the SSM hierarchy (upward and downwards in hierarchy) until either; corresponding sub-queries resolve at the local nodes or the query reaches the root node without any resolution. The SSM model was simulated and its prototype architecture supported by DARPA was developed. The performance of the model was evaluated under various schema distributions, query complexity, and network topology. The simulation results showed that both precise and imprecise queries incur comparable cost, and hence have compatible performance [3]. In certain cases, the SSM imprecise query processing even outperforms a precise query processing. The SSM model provides the following benefits over other multi-database solutions.

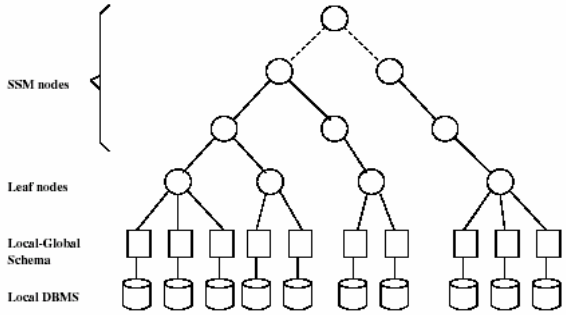

Figure 1: SSM Architecture.

- The SSM provides global accesses to data without requiring precise knowledge of data names or locations,
- The SSM meta-data incurs infinitesimal memory overhead compared to global schema approach.

2.2. Authorization Model for the SSM

In authorization model for SSM, the individual subject in local databases is mapped to a common role defined at Multi-database system (MDBS) level. Consequently, each global access term is tagged with a set of roles allowed to access those objects. The proposed authorization model specifies objects and authorized subjects at global level without violating local autonomy.

At local databases, each local subject accesses local objects according to access control rules defined locally and independently. No further assumptions are made about authorization models used at local databases. At MDBS level, access terms in the hierarchy are populated according to their word relationships. The mapping of individual local subjects to global subjects without violating the local autonomy is a tedious and error prone task. However, use of roles as global subjects simplifies this task considerably. Local databases can now maintain a table that maps subject to its global role. If a new role is added or an existing role is deleted in the global system, all local databases are informed and their local subjects can be remapped to new roles.

2.3. Query processing in enhanced SSM

In our simulated environment, a query is attached a randomly generated matching probability and submitted at an SSM node. The attached matching probability is compared with level probability of SSM node. A successful match results if the matching probability is less than or equal to the level probability. On a successful match the query is transferred to a lower level. An unsuccessful match transfers the query to higher SSM level. A query is accepted when any local database accepts the query.

In enhanced SSM, a randomly generated rank is also assigned to the query. This rank is used to compare with the level rank after a successful match is realized. If the query rank is lower than the node's rank, the query is *rejected* immediately at that node. Thus, a query is either *accepted* at local nodes, *rejected* at SSM root node, or *invalidated* due to insufficient authority at a SSM node.

2.4. Simulation Results

Imposing authorization information to the SSM adds both space and time overhead and clearly affects the query resolution of the SSM. However, the simulation results showed that the authorized SSM offers better performance than the original SSM. This is because the queries with insufficient authority are rejected as early as possible in SSM. This reduces both the workload at each node and the network traffic in the SSM hierarchy. Consequently, the response time of accepted queries is reduced.

3. Extended Authorized SSM

3.1. Wireless Extension to Authorized SSM model

3.1.1. SSM Clients

SSM is currently deployed in a wired environment with clients perceived to be physically connected to the SSM nodes. Extending the scope of SSM to allow mobile clients would overcome limitations posed by a wired environment and would satisfy the need for wireless services to SSM clients. The scope of the SSM can be further expanded by allowing mobility at the local database level. Information origin is transparent to users accessing it and hence, there lies no requirement for information to be accessed only on wired links. Another reason to allow mobile information is due to the rapid growth rate in the range of information accessible to a user at any given time. Hence, we allow local databases to be stationary, mobile, or a mix of both.

3.1.2. Security

Solutions to the wireless extension of the SSM need to consider different aspects and limitations imposed by the unreliable, uncertain, and unpredictable communication link. Security is the primary requirement due to the vulnerable nature of wireless communication link.

We extend the Authorized SSM model in a secure wireless environment with support for mobile clients and mobile/stationary local databases. Initially, we identify the fundamental requirements for a secure transmission. We then propose an integrated solution that covers all identified requirements.

3.2. Security Requirements in Wireless Environment

Security is defined as the science of protecting computers, network resources, and information against unauthorized access, modification, and/or destruction. To provide security in wireless environment, a secure session needs to be setup initially between two parties (a client and base station/access point). The messages transferred after this initial setup needs to be encrypted to prevent any third party from eavesdropping. Also, non-repudiation needs to be prevented. Security thus mainly involves *Confidentiality, Authentication, Integrity checking,* and *Non-repudiation* [6]:

Cryptography has been traditionally studied to provide solutions to such security requirements. A cryptographic system consists of two fundamental components: a complicated mathematical component called an *algorithm*, and one or more secret or public values, called *keys*. Cryptographic algorithms can be further classified as the *secret-key algorithm* and the *public-key algorithm* [5]. We propose a solution using cryptographic algorithms to securely extend SSM model to a wireless environment in the next section.

3.3. Proposed protocol

- Authentication or secure session setup is provided by password based systems where a user is assigned a user-name and password to authenticate in the system. Diffe-Hellman Key Exchange Protocol [6] is used for authentication.
- Confidentiality and Integrity checking is provided by encrypting messages during message exchange. Advanced Encryption Standard (AES) [8] is used for encryption of messages in the wireless network.
- Non-repudiation in secret-key algorithms is possible using Digital Signatures. To decide whether a user has sent a message, he/she includes a digital signature in the message. The digital signature is then decrypted at the receiver using the shared secret key. We provide authorizations through digital signatures (message authentication code MAC) using SHA-256 hashing utilities (SHA-256 is a variation of a 256-bit symmetric block encryption algorithm) [9].

The above algorithms are chosen primarily for two reasons: First, the energy consumed during computation, transmission, and reception of data using these algorithms is already measured and readily available [2]. Second, the AES (Rijndael) and SHA-256 are recognized as standard algorithms by National Institute of Standard and Technology [10] for providing encryption and digital signatures facilities. It is also shown that breaking the Diffie-Hellman protocol is equivalent to computing discrete logarithms under certain assumptions [6]. The details of the protocol using these cryptographic algorithms are given as follows:

- The client initiates the handshake by sending a **client_hello** message to the server. This message contains session id, key refresh rate, private key encryption algorithm and its mode of operation, message authentication code (MAC) algorithm and a random number used to generate the encryption and MAC keys from the master_secret
- The server responds with a **server_hello** message accepting the security association proposed by the client, another random number (to be used during encryption key and MAC key generation), the **server_certificate** for authenticating itself to the client, its share $X = (g^a \bmod n)$ of the master_secret, where X is a large random number (**server_key_exchange**) and a **certificate_request** from the client.
- The client replies with its **client_certificate,** its share $Y = (g^b \bmod n)$ of the master_secret (**client_key_exchange**), where Y is a large random number and a **certificate_verify** message to verify its certificate.
- Finally, the client and the server exchange **change_cipher_spec** message to activate the session with the negotiated security association (the encryption algorithm and its mode of operation, the MAC algorithm, the session id, and the key refresh rate) and a **finished** message to indicate successful key exchange.

Figure 2 summarizes the handshake protocol.

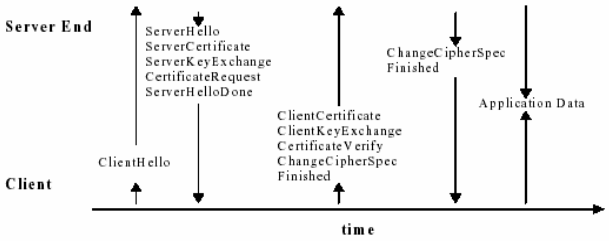

Figure 2: Messages exchanged during the hand-shake protocol.

3.4. Test-Bed

The aforementioned protocol has been implemented using Wireless transport Layer Security (WTLS) in Wireless Application Protocol (WAP). Measurements for energy consumption are taken based on the following test-bed [2].

- PPT2800TM Pocket PC with following features
  - Windows CETM Operating System
  - 32 bit, 206 MHz StrongArmTM SA-1110 processor
  - 16MB flash ROM
  - 16MB RAM
  - 16KB instruction cache
  - 8KB data cache
- 11 Mbps Spectrum® WLAN card based on IEEE 802.11b
  - Operating in Polling Mode P1
- WildcardTM FPGA board from Annapolis Microsystems
  - Supports a Xilinx Virtex XCV300E FPGA
  - 400,000 System gates

- 130,000 block RAM bits (to accelerate cryptographic computation).

4. System Architecture and Design

4.1. System Design

The system comprises mobile clients, servers, and local databases as separate concurrently running processes. The server is a base-station in a cell and also an SSM node in the SSM hierarchy. The connectivity in the SSM infrastructure is through wired links; however, the clients and the local-data sources are connected to this infrastructure through wired/wireless. Local-data sources which are mobile are free to move in any direction. For simplicity, we have ignored the routing time required to locate a local-data base after a query needs to be transmitted from SSM level 1 to a local data source.

The sequence of actions exchanged between a client, an SSM node, and a database is as follows. The clients initially authenticate using the Diffe-Hellman Key Exchange Protocol. If a client is not authenticated, any requests from that client will be dropped by the server. Assuming that the client authentication is successful, the client starts generating requests and transmitting them to the server. The client encrypts requests with advanced encryption algorithm (Rijndael) and appends a message authentication code (MAC) to the data being sent. The MAC will be used in case of non-repudiation. The server acknowledges a request receipt to the client in order to reduce retransmissions. The server or SSM node receiving the request resolves the request and replies immediately in case the request is invalidated or rejected. In case, the query is resolved, it needs to be sent to a local data source. A reply received by the server or SSM node from the database is forwarded to the client. Again, for simplicity, we ignored the time required to locate the client when request response is received by the server and needs to be forwarded to the client.

4.2. Workload and Parameters

The workload of the system consists of imprecise requests (queries) submitted by clients at any node in SSM hierarchy (i.e. to any base-station). In every cell, there are a fixed number of clients (*nclients*) mobile in nature. Mobile clients move across cells and generate requests through different SSM nodes. Normally, every client generates a request (*nrequests*) and waits for its resolution before making further requests. However, it is possible that a client may also generate further requests before previous requests are resolved. A new request is generated after *think_time* - the client waits for *think_time* before generating a new request.

System Parameters related to the generated SSM hierarchy are; number of local nodes, number of levels including the local database level, maximum number of children per SSM node, and level probabilities assigned to each level. System parameters inclusive to the extended authorized SSM model with mobile clients and mobile databases are; SSM (server) vacation percentage, SSM maximum utilization, percentage of clients not registered with the SSM facility, and the percentage of mobile data sources.

We assume that the processing time for generating a request, resolving the request at SSM node, and processing at local databases is exponentially distributed with a mean *proc_time*. A wired/wireless link exists between clients, local data sources, and SSM infrastructure. Also, a wired communication link exists between nodes in SSM hierarchy. The transmission time on a wired link is *comm_time* and on a wireless link is *wireless_comm_time*. We limit the percentage of mobile local databases. Hence, *static_databases* denotes the percentage of static local databases. We also assume that the total wireless bandwidth available is *bandwidth*. The clients send requests only if sufficient bandwidth is available. The available bandwidth channel is symmetric in nature (i.e. Uplink and downlink channels are equally divided). The average *wait_time* in case of insufficient bandwidth is exponentially distributed. The client senses the channel again after waiting an average *wait_time* for insufficient bandwidth. Assuming that every accepted request needs to travel over four types of wireless links (client-server, server- mobile database, mobile database-server and server-client), we assume that the timeout period for request reply at client is at least four times the transmission time over wireless link. The timeout period *time_out* can hence be defined as a period where the client waits for a request response from the server. A client is allowed to retransmit *max_transmit* times. The client drops the queries in case they cannot be completed after *max_transmit* times (i.e. a complete query processing activity includes successful transmission to server, resolution at local database, and successful transmission back to the originator). The server is assumed to accept requests only if it is available. A server is unavailable if it is on a vacation or if it is serving requests at maximum allowable utilization. A server will take vacation for maintenance or in case of emergency during failures. The percentage of total simulation time a server will be on vacation is *server_vacation* (possibly under maintenance, or disconnected due to weak signal, or unavailable due to system failure). Similarly, the maximum allowable server utilization to prevent from problems like thrashing, etc… is *max_utilization.* We also assume the existence of requests generated by invalid users - unregistered clients to the SSM facility defined as *unregistered_clients.* All parameters and their default values are stated in the following Tables 1 and 2.

Table 1: System Parameters

Parameters	Default Values
Number of LDBs	250
Maximum number of children per SSM node	3
Prob. of an invalid query	0.1
Number of ranks	6

Overall, due to mobility, numerous factors like disconnections and unreliability need to be considered. Hence, we expect the query response time in extended authorized SSM platform to be more than the query response time measured in authorized SSM model. However, the network traffic is considerably reduced due to unauthenticated clients, or due to attackers who replay query requests at SSM nodes. Further, the authorized role

based SSM model rejects queries with insufficient authority as early as possible as reported in [4].

Table 2: Workload and System Parameters

Parameters	Default Values
Number of clients in a cell	10
Client think time	2 sec
Total number of SSM levels	7
Number of concurrent requests per user	1-2
Level probability	0.2-1.0
Processing time of a node/ request or of a mobile unit generating a request	0.001
Communication time through wired link	0.003
Communication time through wireless link	0.100
The time a client needs to wait when sufficient bandwidth is not available	½ * wireless_comm_time
The least time a client waits to retransmit a request	10 * wireless_comm_time
Server is down	15%
Server cannot accept requests	90%
Percentage of clients unregistered	10%
Percentage of local databases static in nature	80%
Bandwidth available for transmission	112kbps
Maximum number of Retransmissions	25

5. Experiments and Results

5.1. Experiment 1

This experiment measures the average response time when the same set of imprecise queries is submitted at different SSM levels for different SSM configurations. The number of local databases is kept constant while the height was varied - five, seven, and nine levels. One can conclude from Figure 3 that the response time increases dramatically as the submission level increases. It can also be observed that, regardless of the SSM configuration, the difference in query response times is smaller at lower levels than at higher levels. This is due to the fact that a query submitted at higher SSM levels travels longer than a query submitted at lower level before being accepted, rejected, or invalidated. One can also conclude that SSM configurations with same number of local nodes and lower heights (i.e. bushy structures in nature) would have higher response times than SSM configurations with higher heights (tall and lean structures in nature). This is because of the higher semantic contents at each summary schema node when SSM structure is bushy relative to semantic concentration in a tall and lean structure. As a result, a query made to a particular SSM level would incur more communication links in a bushy structure than in a tall and lean structure before being accepted, rejected, or invalidated.

An extension to experiment 1 was carried out and the average communication cost per query was measured. The communication cost is defined as the number of wired links in SSM that a query traverses before it is being accepted, rejected, or invalidated. Figure 4 depicts the results and as anticipated, the communication cost increases as SSM level number increases. This is because of the longer time (and hence more links) a query spends in the system when submitted at higher SSM levels. Another observation is the exponential nature of the curves which supports the results obtained in Figure 4.

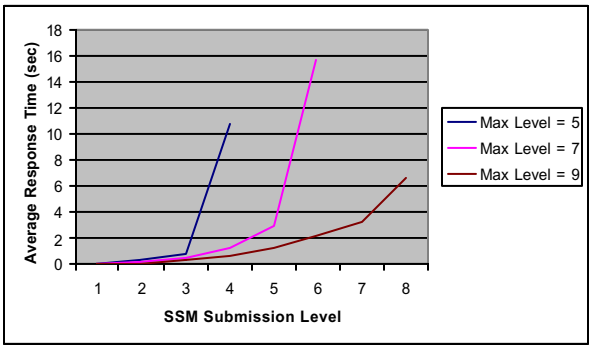

Figure 3: Effect of SSM height on Query Response Time.

In addition, we looked at the completion ratio of client queries across different SSM levels for different SSM configurations. Figure 5 displays the results. As one can conclude the query completion percentage decreases as SSM level increases. This is because of our earlier observation that submission of queries at higher SSM levels results in longer response time and consequently, more frequently a mobile client reaches its *wait time threshold*. In addition, regardless of the SSM configuration, the query completion ratio decreases at slower rate when queries are submitted to lower SSM levels.

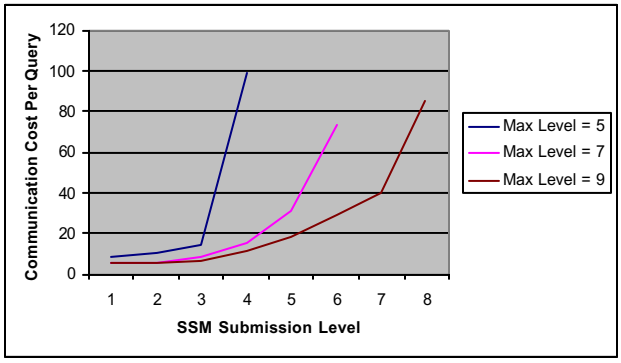

Figure 4: Effect of Total SSM levels on Total Communication Cost.

5.2. Experiment 2

This experiment measures the average response time of imprecise queries for both authorized SSM model and extended authorized SSM with differing percentages of *static_databases* allowing different number of concurrent queries submitted by a user.

As anticipated, the average response time for extended authorized SSM is considerably higher than the authorized model due to the mobility of the clients and local databases. This is due to numerous problems introduced by the mobility - unreliability, disconnection, server unavailability, poor bandwidth, and message garbling due to collisions.

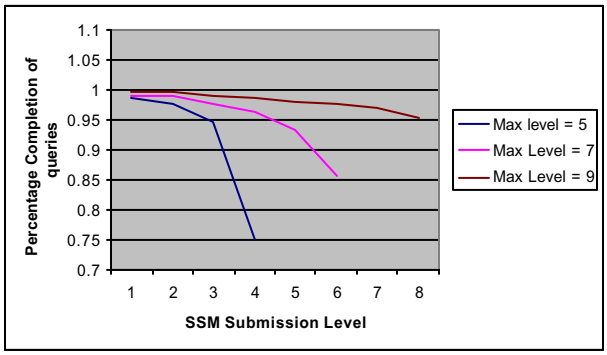

Figure 5: Query Completion Percentage.

From Figure 6 it can be observed that the query response time increases as percentage of mobile databases increases. This is due to the increase in wireless links between the local databases and SSM level 1. As number of mobile local databases increases, the workload of SSM nodes at level 1 increases drastically, resulting in an increase in query response time.

We also performed an experiment allowing users to submit concurrent requests. It was observed that the increase in the number of concurrent requests by a client increases the response time (Figure 7). Again, at lower SSM levels, there is smaller difference in query response time because of the overall system load; as the queries are submitted to the higher SSM level, the load in the system increases resulting in a higher network traffic and hence an increase in response time.

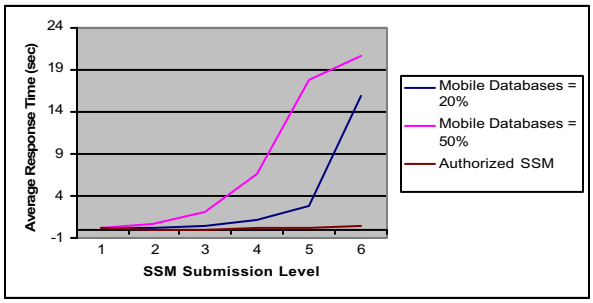

Figure 6: Query response time.

5.3. Experiment 3

It was noted earlier that the server availability is a factor that would contribute largely to query response time. In this experiment, we study the effect of server unavailability on the query response time. Server unavailability is due to factors like insufficient bandwidth, disconnections due to weak signals, server maintenance, server failures, and server utilization above threshold. Hence, server availability is defined as the window between the threshold for server vacation and the threshold for maximum allowable utilization.

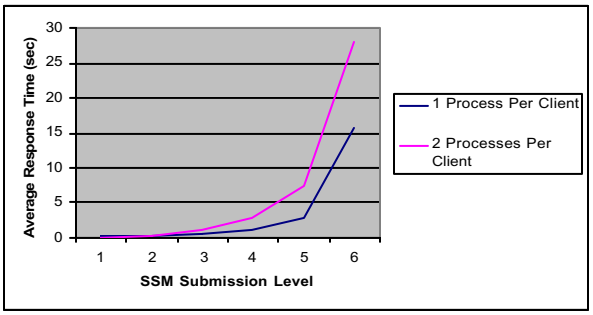

Figure 7: Effect of number of concurrent requests at SSM nodes (80% static databases)**.**

In this simulation run, the server maximum allowable utilization is varied from 75% to 95% in steps of 5%. The server vacation is also varied from 5% to 25% again in steps of 5% (Figure 8). Based on different combinations of server vacation and maximum allowable server utilization, the query response time is measured at SSM level four for one process per client. Interestingly, lower server availability, results in lower query response time. This is because more clients are unable to connect to SSM due to poor availability of SSM. This reduces the traffic in the network and hence, reduced query response time.

In the above experience, we also measured the number of clients unable to connect to the SSM infrastructure. It was observed that as the server availability window increases, the number of clients unable to connect decreases (Figure 9).

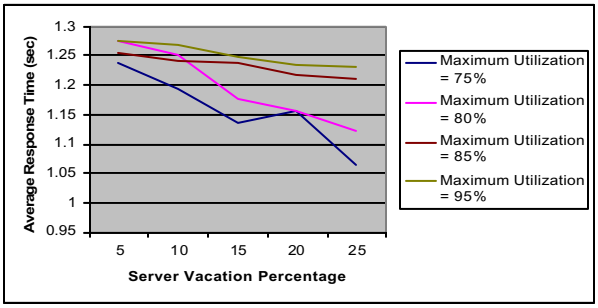

Figure 8: Effect of server unavailability on query response time.

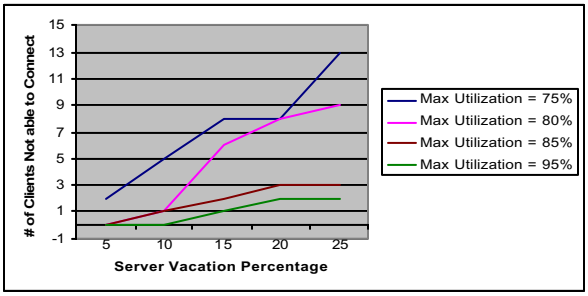

Figure 9: Effect of server unavailability on clients not able to connect.

5.4. Experiment 4

In this experiment, we vary the percentage of unregistered clients. The same set of queries is submitted to different SSM levels, and the query response time is observed. Figure 10 shows our observations. One can

conclude from Figure 10 that the query response time decreases as the percentage of unregistered clients increases. This is because the SSM node blocks unregistered users from utilizing SSM facility, as early as possible, thus ensuring lesser traffic in SSM hierarchy. We assume that at least 50% percent of the clients are always registered with SSM facility. In this experiment we vary the unregistered client percentage in steps of 5% from 5% to 30%.

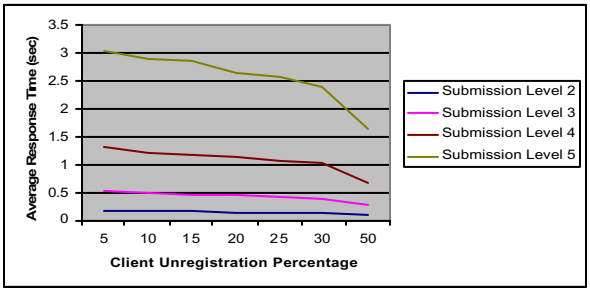

Figure 10: Query response time *vs.* Unregistered Client.

5.5. Experiment 5

In this experiment, we measure the energy consumed when the same set of queries are submitted across SSM levels. The energy consumption was separated into different categories as follows:

- Total Energy consumed during **Session Refreshing** after every transmission of 2MB of information by the client,
- Total Energy consumed during **Key Refreshing** after every transmission of 128KB of information by the client,
- Total Energy consumed during **Transmission** of data over wireless link,
- Total Energy consumed during **Retransmission** of data when there was no acknowledgement received due to limitations of wireless medium and server unavailability,
- Total Energy consumed in **Idle mode** between transmissions.

Initially, the server availability window was kept constant and the total energy consumed was measured across different SSM levels. The total energy consumed increases exponentially (Figure 11) as the SSM submission level increases. Additionally, we also measured the session refresh energy, the key refresh energy, the transmission energy, the retransmission energy, and the energy consumption during the idle operational mode of mobile clients separately. We observed that the key refresh energy, transmission energy, and retransmission energy remained approximately the same across SSM submission levels (Figures 12 & 13) - due to the constant data exchange between the client and server. Thus, increase in total energy is primarily due to two factors: increase in (i) session refresh key energy and (ii) the idle mode energy. From Figure 12 it can be observed that there is a drastic change in power consumption when queries are submitted at level 2 and level 3. This is because of the dramatic increase in the number of sessions due to the hierarchical nature of the SSM.

Energy consumption due to the idle operational mode is the second factor governing the total energy consumption across SSM submission levels. We observed that idle mode energy consumption increases as SSM level increases. This is because of longer response time as experience earlier (see Figure 3). As a result, the mobile client device remains idle for a longer time waiting for a resolution.

We changed the server availability window in the second part of this experiment and measured the total energy consumption across different SSM submission levels. We observed the power consumption at level 4 for server vacation 5% while varying server maximum utilization. As seen in Figure 14, the power consumption reduces as the server availability increases. The two factors that govern the power consumption are session refresh energy and the retransmission energy. Increase in the server availability window reduces the number of dropped queries and hence, decrease in number of query retransmissions. This accounts for the decrease in the retransmission energy, and hence total energy.

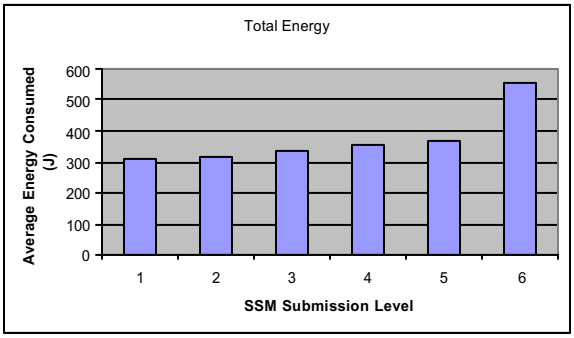

Figure 11: Total Energy.

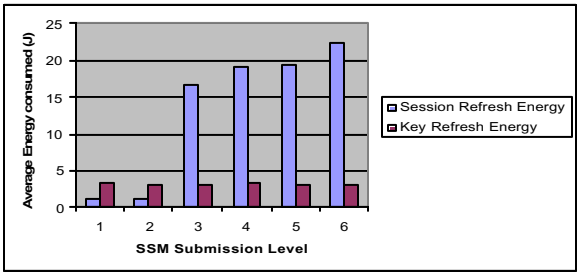

Figure 12: Session Refresh Energy, Key Refresh Energy Consumed.

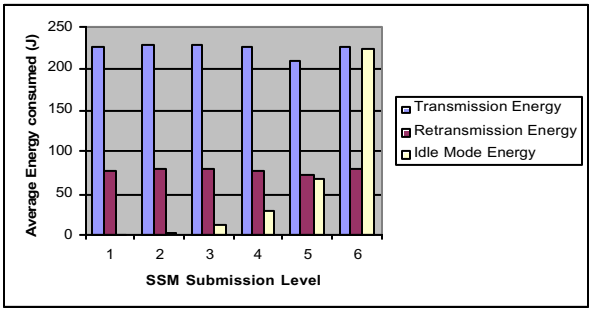

Figure 13: Transmission, Retransmission, Idle Mode Energy Consumed.

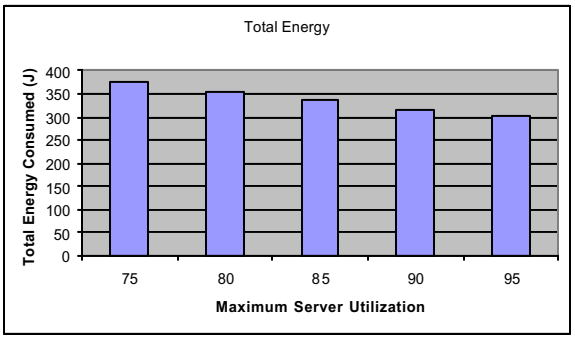

Figure 14: Total Energy Consumed different Server Availability Windows.

Figure 15 displays the change in session refresh energy and retransmission energy that affect the change in total energy.

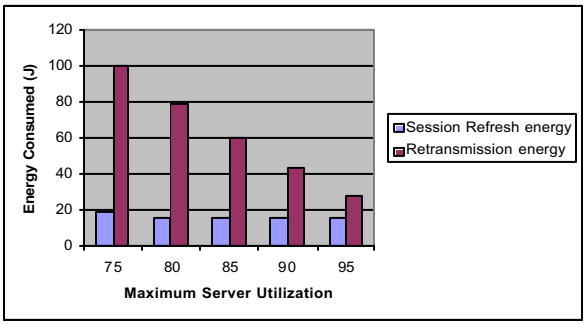

Figure 15: Session Refresh, Retransmission Energy consumed.

6. Conclusion

Enforcing security in multi-databases was identified in [4] as an important issue. However, when these secure applications are extended to wireless environments, the security is compromised due to factors introduced anew by the wireless nature of the environment as such; denial of service, eavesdropping, unreliability of communication medium, frequent disconnections, insufficient bandwidth, and server unavailability, etc.

In this paper, we successfully extended a global information processing system (Authorization model of SSM) to a wireless environment. We identified the fundamental security issues and sought a solution satisfying them in a wireless environment using various cryptography algorithms. We have simulated a secure wireless environment and carried out experiments to study the changes in response times and power consumption. It was shown that:

- A bushy SSM configuration (SSM configurations with lower heights) had higher query response times than a tall and lean SSM configuration (SSM configurations with higher heights),
- The query completion ratio drops as SSM submission level increases,
- More the number of mobile databases, longer will be the response time (due to the additional wireless links that need to be traversed),
- Smaller the server availability window (i.e. higher vacation and lower utilization), smaller is the response time (due to more number of queries being dropped),
- As the number of concurrent requests per client increases, the response time also increases. However, the increase in response time is significant at higher levels and marginal at lower levels,
- As percentage of clients registered to the SSM facility increases, the response time decreases (due to the less number of network traffic in SSM hierarchy and local database links),
- The energy consumption increases exponentially as SSM submission level increases. The energy consumption is governed across SSM submission levels by the idle mode energy consumption and the session refresh energy consumption.

References

[1] J.B. Lim and A. Hurson "Heterogeneous Data Access in a Mobile Environment" in *Advances in Computers*, 48: 257-314, 1999.

[2] Ramesh Karri, Piyush Mishra "Minimizing the secure wireless session energy" Journal of Mobile Network and Applications (MONET), Spring 2002

[3] Bright, M.W., Hurson, A.R. and Pakzad, S. "Automated Resolution of Semantic Heterogeneity in Multidatabases," ACM Transactions on Database Systems, 19 (2): 212-253, 1994.

[4] S Ngamsuriyaroj, A.R. Hurson, T.F. Keefe "Authorization Model for Summary Schema Model", International Database Engineering and Applications Symposium, 2002, pp. 183-191.

[5] James D. Solomon, Mobile IP the Internet Unplugged. Prentice Hall PTR; 1st edition (January 15, 1998)

[6] Diffie-Hellman key exchange protocol: http://modular.fas.harvard.edu/edu/Fall2001/124/lectures/lecture8/html/node2.html

[7] J. B. Lim, A. R. Hurson "Transaction Processing in Mobile, Heterogeneous Database Systems" trans. on Knowledge and Data Eng., 14 (6): 1330-1346, 2002.

[8] AES Encryption: http://csrc.nist.gov/encryption/aes

[9] http://csrc.nist.gov/encryption/tkhash.html

[10] http://www.securitytechnet.com/rsc-center/link.html

[11] D. Jonscher, Klaus R. Dittrich: An Approach for Building Secure Database Federations. In Proceedings of the VLDB Conference, Santiago, pp. 24-35, 1994.

[12] Ching-Yi Wang and David L. Spooner "Access Control in a Heterogeneous Distributed Database Management System", IEEE 6th Symp. on Reliability in Distributed Software and Database Systems, pp. 84-92, 1987.