CHAPTER 1

INTRODUCTION

1.1 BACKGROUND

The main objective of Database Management System (DBMS) is to provide an environment that is both convenient and efficient for the commercial organizations to store and access the stored information. Relational database management systems are designed to maintain large volumes of data related to an enterprise. The data related to an organization are stored in the form of tables in relational databases. It provides Data Definition Language (DDL) and Data Manipulation Language (DML) commands which are used to create the structure of data, insertion, deletion and manipulation of data. Due to the constraints associated with the relational database management system, it is not possible to accommodate the data in simple naïve user understandable format. The strict rules, procedures and constraints must be followed to store and manipulate the data as specified in the relational database management systems. It is difficult to exchange the data between different real-world applications in the internet environment using relational database management systems. Data must be transformed into a common format before it is exchanged between the applications.

Data used in internet applications and commercial business applications can be either structured or semistructured or unstructured. Structured data can be easily represented in either traditional Relational Database Management Systems (RDBMS) or Object Oriented Database
Management Systems (OODBMS) or File Systems. It is difficult to manage unstructured or semistructured data. The data which does not have a rigid structure or structure can be frequently changed is called semistructured data. Separate tools or generalized software is required to manage unstructured or semistructured data. Most of the commercial database management systems are platform dependent and proprietary software. Extensible Markup Language (XML) (Bray et al 2000) becomes a standard data format for the management, display and organization of data in a platform and language independent way to represent both unstructured and semistructured data. Initially it was not designed for database applications. The syntax for XML is derived from hypertext markup language and Standard Generalized Markup Language (SGML). The World Wide Web (WWW) is based on Hyper Text Markup Language (HTML). In hypertext markup languages each tag has its predefined meaning and its purpose. User can define his own tag in XML for data descriptions. XML is an essential technology for the user working with data whether on the web or internally. It is a technology concerned with the description and structuring of data.

An XML document can be modeled as an ordered labeled tree. Each node in the tree to represent an element or a value. Values are represented by character data (CDATA) or Parsed Character Data (PCDATA) and occur at the leaf nodes. The tree edges represent a relationship between two elements or between an element and a value such as parent, child or ancestor / descendent relationships. Each element can have a list of (attribute, value) pairs associated with it. Attributes and elements are considered as the same without any differences between them.

During the last decade, the database researchers focused their efforts towards the efficient manipulation of XML documents. Searching all occurrences of a query pattern in the XML document is one of the important
operations. There are several demerits associated with the data retrieved from the XML document if XML documents are stored in relational databases using queries. The data stored in the XML document are accessed with the pattern matching methods such as TwigStack (Bruno et al 2002) algorithms or ViST (Wang et al 2003).

The optimal XML pattern matching algorithm proposed by Bruno et al (2002) is a stack based. It processes input lists of element instances for tags that appear in a query tag. This algorithm operates on the positional representation of elements to identify the matches. The improved version of TwigStack used the XB-trees to increase the speed of the processing input lists. The XB-trees discard the sections of input without losing any matches. The main drawback of Twigstack is the elimination of section of data depending on the distribution of the matches. If the matches are prevailed all over the data list, it takes more time to search the data including leaves to eliminate missing matches. Another disadvantage is that it suffers from sub optimality for parent-child or ancestor-descent relationship in a query.

The method presented by Wang et al (2003) was called as ViST that transforms the XML document in the form of hierarchical tree and input queries into structured encoded sequences. In this approach, the structured encoded sequences are represented in the form of two dimensional sequences of (symbol, prefix) pairs \{(a_1,p_1),(a_2,p_2),\ldots(a_n,p_n)\} where \(a_i\) represents a node in the XML document tree and \(p_i\) represents the path from the root node to node \(a_i\). The nodes are in pre order. The transformed sequences are stored on disk with the help of B^+ trees. The inefficiency of this approach is that it takes much storage space.

Rao and Moon (2006) presented a framework to efficiently transform the XML documents to sequences and process the twig queries without breaking into root-to-leaf path and process them individually. But the
transformation of XML into sequences takes more time and sequences are stored in relational databases. The existing standard indexing techniques are also adapted into the sequences stored in the form of tables. Now the queries in the form of sequences are issued against the XML data in the form of sequences in tables. This transformation operation and sequence matching for data retrieval are time consuming tasks.

Version management methods for XML documents are essential in the management of web based information and collaborative work environment. The earlier techniques for version control techniques such as Revision Control System (RCS) (Tichy 1985) and Source Code Control System (SCCS) (Rochkind 1975) are edit based representations of managing the multiple versions of XML data. The version scheme proposed by Chien et al (2001) is based on temporal clustering with editing based representation. User specified threshold must be specified for creation of versions.

1.2 RESEARCH MOTIVATION

The majority of the clustering algorithms for XML documents available in the literature (Lian et al 2004; Dalamagas et al 2006; Nayak and Iryadi 2007) are used to perform the inter document clustering only. In order to improve the efficiency of data retrieval, it is essential to cluster the elements present in the document itself that is to perform intra document clustering. The existing standard indexing techniques for XML documents in relational databases are inadequate to retrieve the required data stored in flat files. The clustering with an indexing combined approach for XML data handling is not adequately addressed by the available literatures.

Version management is one of the challenging research areas in XML databases. In software configuration management, software installation, collaborative work environment and software development, version
management techniques are essential to manage different versions of code or data. It needs to keep track of the changes made to the software or data. The majority existing version management techniques are used to create and manage the versions of XML document based on the change detection algorithms. The data in the form of objects in object oriented database or in flat file require a model to create versions based on the importance of attributes modification and mapped into XML data. These factors motivated to work towards proposing new efficient techniques for XML data management. The overall architecture model for this research work is given in Figure 1.1.

Security is very much essential for the web based applications which exchanges data between them through internet. The performance of web application is one of the most important criteria that must be provided to meet the user requirement. Performance degradation is not acceptable when encryption and decryption techniques are used for providing security. Public key based cryptography algorithm such as RSA is most suitable for exchanging XML data securely over the web. This method provides confidentiality, authentication and integrity. The performance of this public-key cryptography used for XML data exchange is analyzed with the implementation of RSA (Rivest et al 1978) algorithm in Redhat Linux and Windows XP platforms.

1.3 LITERATURE SURVEY

Extensive research work started on XML data representation, manipulation, and presentation a decade ago because of its widespread usage. In this chapter, a survey of related works on indexing, clustering, versioning, Public key encryption and XML databases are considered.
1.3.1 Related Works on Clustering

Earliest clustering algorithm K-means (MacQueen 1967) is one of the simplest unsupervised learning algorithms that solved the well known clustering problem. It is an easy method to classify a given data set through a specified number of clusters. The focus is to define k centroids, one for each cluster. This algorithm is sensitive to the initially selected center and does not
provide an optimum configuration solution. It may be invoked for a multiple
times to generate the optimum solution.

Fuzzy c-means (FCM) (Dunn 1973) is a method of clustering which allows the same data item to belong to two or more clusters. This algorithm has been improved by Bezdek (1981). It is frequently used in pattern recognition and pattern matching.

Zhang et al (1996) developed a data clustering method named Balanced Iterative Reducing and Clustering using Hierarchies (BIRCH), and demonstrated that it is especially suitable for very large databases. BIRCH incrementally and dynamically cluster the incoming multi-dimensional metric data points to try to produce the best quality clustering with the available resources. This method can typically find a good clustering with a single scan of the data, and improve the quality further with a few additional scans. BIRCH is also the first clustering algorithm proposed in the database area to handle "noise" (data points that are not part of the underlying pattern) effectively.

Steinbach et al (2000) reported the performance of clustering techniques such as Agglomerative hierarchical clustering and K-means. Two categories of k-means clustering techniques such as standard k-means and bisecting k-means are experimentally analyzed. The experimental results show that bisecting k-means performs better than the standard k-means techniques. Hierarchical clustering approach is a quality clustering method but it takes more time to form the clusters. K-means taking a linear amount of time with respect to the number of documents. It provides poor quality clusters. Hybrid approach formed by combining hierarchical agglomerative and K-means is suitable for the document browsing system. Hierarchical clustering provides quality and k-means is used for run time efficiency.
Bitmap indexing based clustering technique was recommended by Yoon et al (2001). Based on the similarity and popularity operation in bitmap indexes, three clustering techniques such as top-down clustering, bottom-up clustering and mixed clustering were proposed. The data retrieval performance is improved with this approach.

Lin et al (2001) presented an algorithm based on similarity to cluster the documents. Randomization is used to make the clustering more efficient. For the given two documents x and y there is a function \( 0 \leq f(x,y) \leq 1 \) which returns to show how similar x and y are. It takes K as a parameter to specify the number of clusters expected.

A lot of efforts have been taken on how to cluster XML documents effectively with structural (Nierman et al 2002; Leung et al 2005; Dalamagas et al 2006; Hwang et al 2010) or semantic (Lee et al 2001; Tagarelli and Greco 2004; Nayak and Iryadi 2007; Kim et al 2008) information.

The structural similarity between a pair of XML documents can thus be computed based on different edit distances (Andrew and Jagadish 2002) which differ from each other in terms of the set of allowed edit operators and their support for repetitive and optional XML elements. It has been proved by Zhang et al (1992) that computing the edit distance for unordered labeled trees is NP-Complete and yet the distance is not optimal in any sense related to the elements semantics.

Linked XML documents are clustered by an approach (Catania and Maddalena 2002) by including the hypertext link. Graph based formalization is proposed for hypertext link and compute the number and the type of links connected to the documents. A model is proposed for different types of link representation.
Distance based clustering of XML documents (Francesca et al 2003) focus on the notion of XML cluster representative. It is a prototype XML document containing the most relevant features of the set of XML documents within the cluster.

Dynamic XML documents clustering (Rusu and Taniar 2008) are established as an intelligent and efficient technique to reassess the distance between dynamic XML documents when one or all of the initially clustered documents have changed. It allows the user to reassess the pairwise XML document distances, not fully comparing each new pair of versions in the clustering solution, but by determining the effect of temporal changes on the previously known distances between them. It is both time and I/O effective, as the number of operations involved in distance reassessing is reduced.

Cheng et al (2003) implemented an efficient clustering based indexing method for long path expression queries. It proved that the existing inverted index and join algorithms provide better performance in short path expression queries. Clustering concept is used to enhance the inverted index. Prototype model of XML database was developed to process the path queries.

The Hierarchical algorithm presented by Lian et al (2004) is called S-Grace for clustering XML documents based on structural information present in the data. The notion structure graph (S-graph) is specified, supporting a computationally efficient distance metric defined between documents and set of documents. The structure graph represents XML data and distance metric is used to perform clustering. The path expression specified in the query of this method does not support the wildcard (“*”, and “/”) (ancestor / descendant relationships). The simplest metric yields new clustering algorithm, which is efficient and effective when compared to other approaches.
Principal component analysis based clustering (Liu et al 2004) for XML documents support the same Document Type Definition (DTD). The features extracted from documents are transformed into vectors in high dimensional Euclidean space. The dimensionality is reduced by the principal component analysis.

A clustering based on path pattern is presented (Leung et al 2005). It is a method of XML structural representation called Common XPath (CXP), which encodes the frequently occurring elements with the hierarchical information and proposed to take the CXPs mined to form the some of the feature vectors for XML document clustering.

A clustering model (Yang et al 2005) is developed for representing XML documents. The term semantics, element similarity, as well as elements’s relative importance of a given set of documents can all be taken into account. It has also formulated an iterative estimation procedure for automatically learning an element similarity matrix associated to this model.

Nayak (2006) investigated the role of semantic measures in XML clustering. The influence of semantic computation in semistructured data used for clustering is identified and also the effect of relationship in the form of similarity at individual element is reported. The conclusion shows that there is no significant influence of semantic measures in clustering methodologies.

Semantic search based on clustering, developed by Li (2008) taking into the account of semantic relationship between the results. After clustering the relationship between the given query, the clusters are analyzed and based on that clusters are filtered to remove the irrelevant results. The method is improved by bipartie graph and feature vector matrix. Adjacency list is used to store the bipartie graph.
Yuan et al (2008) proposed a scheme based on the path features. The element tags and their position in the hierarchy of documents are considered for clustering. All path lengths less than or equal to as a feature vector for an XML document is computed. Based on the feature vector matrix, the path based method is improved.

A clustering algorithm for XML based tree model data type was presented by Zhou et al (2008). It focused on integrating data types and tags of XML documents. This approach correctly clustered the real world data set with semantic meaning, even though it contained different tags but described the same object.

Tran et al (2008), presented a clustering model that combines the structure and content of the document. The experiments prove that the clustering of the text centric XML documents, based on the content only information, produces a better result in a homogeneous environment. When the structure and content of the document are combined, for clustering in a heterogeneous environment produces better results.

Nagwani et al (2010) presented a novel method to a cluster collection of homogeneous XML documents by using the weighted similarities on attributes. The weighted similarities are used as a measure for detecting the similarity between the documents. It combines the features of tree edit distances, cosine and jaccord features. This approach requires expensive operations to calculate the document distances.

### 1.3.2 Related Works on XML Indexing Techniques

The wide usage of XML in the recent days leads to represent data from multiple sources. So XML is the format for which data from different sources are integrated. XML data are stored as text or as string format. In this
section, some of the existing indexing techniques available in the literature are presented.

Abiteboul et al (1993) suggested a model for storing and updating of data in files. This model gives poor performance when there is a large volume of modification. McHugh et al (1997) developed a database system called as Lore to store the semistructured data.

The problem of incremental dynamic updating of inverted list was addressed by using the dual structure index by Tomasic et al (1994). This scheme separates the long and short inverted lists and optimizes the updates, storage and retrieval of lists.

Goldman and Widom (1997) presented a system called DataGuide for computing concise and accurate structural summaries of semistructured databases. As far as the semistructured databases are concerned there is no schema fixed in advance and the structural summaries are needed for query formulation and query optimization. DataGuide provides dynamic schemas, created from the database; their support for browsing database structure, formulating queries, storing information such as statistics and sample values, and support query optimization. The theoretical foundations of DataGuide along with algorithms for their creation and incremental maintenance are presented. The implementation of DataGuide in the Lore DBMS for semistructured data provides significant level of query optimization. User interface is implemented along with DataGuide to enable structure browsing and query formulation, and as a means of guiding the query processor and optimizing query execution.

Goldman et al (1998) developed a prototype model for data retrieval from databases based on the proximity of the keywords. In this model, database is viewed as a graph with data in vertices and relationships
are indicated by edges. Proximity is based on the shortest path between objects. Compact indexes are used to find the shortest path quickly.

A novel index structure called Template Index as T-index was suggested by Suciu and Milo (1999) for evaluating query path expression in semistructured databases. It captures the partial knowledge about the structure of the data and the type of queries in the query mixes is represented in the template path. The main features are that it takes less space and T-indexes are efficiently constructed with the help of simulation or bisimulation relation for which efficient algorithms exist. T-indexes are associated with the singular regular expression, which occupy the storage area, which is proportional to the size of the database. In can be considered an elegant index structure with the other existing index structures but index maintenance problems are not addressed.

Luk et al (2000) surveyed the search engines used in XML documents. They analyzed the search engines based on database-oriented approach, information retrieval-oriented approach and hybrid approach. The advantages of the database approach are that (1) data relations, constraints and integrity can be modeled and checked, (2) the query languages are similar to standard database query language (SQL) so that little training is needed to use the system and (3) standard (relational and object-oriented) database engines can be used. Directly importing XML documents to the database engine can be difficult due to the heterogeneous structures of different documents.

In the information retrieval-oriented approach, search based idea on proximity can be applied to XML documents to achieve better precision. The advantages of this approach are that (1) existing information retrieval engine can readily be modified for this purpose; (2) no training is required since it can be used similar to a standard search engine, and (3) it has a smaller
indexing cost since it does not have to contain detailed information about the structures. The problem with this approach is that it may not be as precise as the previous approach because it is based on an approximate matching technique, which does not support the sophisticated matching of document structures. If the user is not interested in uniformly structured results and the user only needs the top $N$ documents, then this approach is adequate. In this survey, there is little work on extending the Boolean model directly for searching of XML documents. In the hybrid approach, some popular techniques are combined, in the search for the relevant Xpaths, for example expressed in XQL, and full text search. This is likely to give a more precise search results since Xpaths specification limits the possible matches. Also, this approach is flexible: working like a standard information retrieval engine or working like a database engine.

Florescu et al (2000) proposed an approach to enhance the capabilities of XML query languages to support the keyword search at the lowest level of XML elements. They implemented their approach on top of the commercial Relational Database Management Systems. Different methods of implementation and results are reported. Keyword search and overall XML query processing can be done efficiently. In this combined keyword search and query processing approach, users have no or only need partial knowledge about the structure.

Cooper et al (2001) implemented a fast index structure on top of relational database management systems to access the semistructured data. This method encodes paths as strings, and inserts those strings into a special index that is highly optimized for long and complex keys. The nesting structure of XML data is viewed as a tree. Two categories of paths are specified such as raw path and refined path. A raw path exists in the data to accelerate any adhoc query to traverse the root-to-leaf of the tree. Refined
path is generated from reorganized portion of the data for frequently occurring and important access pattern. Refined path support the complex queries in a single index lookup with the traversal of the tree from siblings to siblings.

The efficient storage structure and indexing schemes have been developed that consolidate an XML document so that these difficulties can be overcome with Suciu and Milo (1999), Goldman and Widom (1997), Cooper et al (2001), Li and Moon (2001). The concepts and techniques specified in this research work are used for the support of step evaluation along the child and descendent-or-self axes. It is not adequate support for XPath language. These techniques depend on query processing techniques, which call for implementation techniques that lie outside the relational domain.

Navapro (2001) did the survey of techniques to address the problems of string matching allowing errors. This is a core issue for emerging areas such as information retrieval and computational biology. It focuses on online searching and mostly on edit distance, describing the problem and its relevance, its statistical behavior, its history and current development, and the core idea of the algorithm and their complexities. A number of experiments were carried out to test the performance of different algorithms and the best method for each case was reported.

XPath (Berglund et al 2001) is a standard specified for accessing XML data, accepted and approved by the World Wide Web consortium and adapted by all major software developers.

Zhang et al (2001) developed a subsystem for retrieval of XML data from the RDBMS. The queries used to access the XML data are called containment queries. Queries in the form of containment are supported by inverted list method. It can be implemented using (i) in a separate loosely
coupled Information Retrieval Engine, or (ii) using the native tables and query execution engine of the RDBMS. The matured RDBMS technology supports query optimization, query execution, scalability, and concurrency control and recovery. It is difficult to extend immediately the queries and structures that implement these new operations. The performance impact of both options using native implementations in two commercial relational database management systems and in a special purpose inverted list engine is reported. This performance study shows that while RDBMS are generally poorly suited for such queries, under certain conditions they can outperform an inverted list engine. Further analysis identifies two significant effects that differentiate the performance of the information retrieval and RDBMS implementations. They are the join algorithms employed and the hardware cache utilization. The results suggest that contrary to most expectations, with some modifications, a native implementation in an RDBMS can support this class of query much more efficiently.

Li and Moon (2001) presented a model for indexing and querying of XML data. Hristidis et al (2003) developed a system called Xkeyword which provides efficient keyword proximity queries on very large XML database graph. It is built on relational databases. It supports very large graphs and execution of the query is optimized by graph schema. A set of keyword indices is built along with indexed path relations that specify particular patterns of paths in the graph. The query evaluation stage plans are developed with a near optimal set of path relations to efficiently locate the keyword query results. The results are presented graphically using the novel idea of interactive result graphs.

Schmidt et al (2001) presented an efficient approach to retrieve the data from XML database based on the concept of lowest common ancestor of
nodes in the XML syntax tree. In this approach, the end user need not require the knowledge of tags and hierarchies.

Fuhr and Grobohann (2001) developed a query language XIRQL that supports the features of weighting, ranking, relevance-oriented search and semantic relativism with the idea of logic based probabilistic model. The path algebra is also given as a starting point for query optimization.

A search engine to retrieve the XML data efficiently based on the rich annotation and additional information provided by XML and their elements is proposed by Theobald and Weikum (2002). It supports relevant ranking and path queries with wildcard and semantic similarity search condition. It is fully implemented as a suite of JAVA servlets. Tatarinov et al (2002) proposed a framework model to store the ordered XML data in relational databases efficiently by encoding the order as a data value. Using the same encoding method ordered XPath expressions are translated into SQL expression.

A survey work on indexing and searching of XML document was done by Luk et al (2002). The indexing techniques were analyzed based on the category of flat file, semistructured, and structured methods. The full-text search and multistage search techniques are reviewed. The various retrieval models are discussed with the extension for XML data retrieval possibility.

It is essential to approximately compute the sizes of query results and intermediate results for better query optimization and also estimation of the total result size for query refinement at the user level. Wu et al (2002) proposed a novel technique to estimate the query size optimally in an XML database. XML queries more often specify the structural patterns, require the specific relationship between selected elements. The existing techniques can compute the number of nodes that will satisfy a node-specific predicate in the
query pattern, such computations cannot easily be combined to provide the estimates for the entire query pattern, because element occurrences are expected to have a high correlation. The solution to overcome the constraint is based on the histogram encoding of element occurrence position. With the help of such histogram it is possible to compute the size of complex pattern queries, as well as for simple intermediate patterns that may be computed in alternative query plans by means of a position histogram algorithm proposed. The technique is extended to exploit schema information regarding the allowable structure through the use of a coverage histogram. The experimental results for real and synthetic data set prove the effectiveness of this technique.

Grust (2002) suggested a model for a database index structure designed to support the evaluation of XPath queries. It is capable to start traversal from an arbitrary node in an XML document with the additional feature of embedding in Xquery (Boag et al 2002) expression. The new index can be implemented and queried using relational techniques with the support of R-trees.

Hristidis and Papakonstantinou (2002) developed a framework model called DISCOVER which allows the user to retrieve the data from the databases using keyword queries without any knowledge of the database structure or SQL. It finds the results without redundancy. The modified greedy algorithm is adopted in this model.

Gottlob et al (2002) proposed an algorithm for processing XPath queries to improve the efficiency of the present and future XPath engines. The experimental analysis shows that the worst case complexity of access time of XPath queries is exponential to the size of the queries. The main memory based algorithm uses a bottom up approach which means to process the query of the given path traversing the tree from the leaves up to its root.
The bottom algorithm is mapped into top down with modification mechanism. The top down algorithm generates minimum useless intermediate results. Linear time algorithm is specified for processing XPath queries.

Schmidt et al (2002) suggested a framework model for XML data management called XMark. This framework model is used to assess the qualities of an XML database with a broad range of query types, which are encountered in real world experiments. It is very useful for database researchers, implementers and users comparing the XML databases in a standardized application. The sample set of queries is offered where each query is intended to challenge the query processor in a particular context. The overall workload estimation mechanism consists of a scalable and concise database and comprehensive set of queries which include the major context of XML query processing that covers the textual features, data analysis and adhoc queries. The experimental results are reported to the user as a benchmark to compare their results in several platforms. Update specifications don't present in this framework model due to the non-availability of the World Wide Web Consortium (W3C) standards.

Qun et al (2003) implemented a new indexing scheme called D(k) index based on the bisimilarity concept for graph structured documents. This approach possesses the adaptive nature to adjust the structure according to the query workload and support efficient update algorithms.

The framework model that returns the best ranked results from hyper linked XML documents for the keyword search query was developed by Guo et al (2003). The algorithm designed for the ranking of hyper linked XML document includes inverted index list structures and associated query processing module for processing keyword search queries.
ViST (Wang et al 2003), a novel index structure for searching an XML document represents both XML document and XML queries in structure encoded sequence that querying XML data is equivalent to find sub-sequence matches. It provides a unified index on both content and structure of the XML documents.

XISS/R (Philip et al 2003) system is an implementation of the XML Indexing and Storage System (XISS) on top of a relational database. It is based on the XISS extended preorder numbering scheme; it captures the nesting structure of XML data and provides the opportunity for storage and query processing independent of the particular structure of the data. The system includes a web based user interface, which enables stored document to be queried via XPath.

Range-based labeling scheme (Xing and Tseng 2004) is an efficient method to determine the ancestor relationship between two nodes in constant time. In this approach, prefix based labeling scheme is used to support the arbitrary insertion without relabeling of nodes in an XML document.

Grust et al (2004) presented a database index structure called XPath accelerator that has been specifically designed to support the evaluation of XPath path expressions. This index is capable of supporting all XPath axes (including ancestor, following, preceding-sibling, descendant-or-self, etc.). The index has been designed with the XPath semantics. It can be supported well by the existing relational database query processing technology: the index (a) permits set-oriented (or, rather, sequence-oriented) path evaluation and (b) can be implemented and queried using well-established relational index structures, notably B-trees and R-trees. The XPath accelerator index can implement on top of different database backend and the results show that the index performs well on all levels of the memory hierarchy, including disk-based and main memory based database systems.
A node labeling scheme called ORDPATH (O’Neil et al 2004) allows capturing document order and document hierarchy within a single XML column of the primary XML index.

Indexing XML data stored in a relational database (Pal et al 2004) reports a method for indexing XML data in Microsoft SQL server. It discusses the techniques used in Microsoft SQL server 2005 and Microsoft SQL Server 2008 for indexing XML Binary Large Objects (BLOBs). A shredded representation confirming to Infoset items (Cowan et al 2001) of nodes is stored in a B’ tree. This is known as a primary XML index.

Sequencing of tree structures for XML indexing (Wang and Meng 2005) aims at avoiding expensive join operations in query processing. It transforms structured XML data into sequences so that a structured query can be answered holistically through subsequence matching.

Ahn et al (2005) surveyed three types of indexing techniques used for XML data manipulation. Based on the properties they categorized the indexing techniques into sequence based indexes, structural indexes, numbering based index and keyword based indexes. They explored A(k) index, XISS and XISS/R and XPath accelerator and reported the implementation details.

Xu and Papakonstantinou (2005, 2008) in their research work, recommended an efficient algorithm based on the notion of lowest common ancestors called Indexed Stack to find answers to keyword queries based on XRank's semantics to Lowest Common Ancestors. The results show that the Indexed Stack algorithm outperforms in terms of both Central Processing Unit (CPU) and I/O costs other algorithms by orders of magnitude when the query contains at least one low frequency keyword along with high frequency keywords.
Lian et al (2005) presented an efficient framework model for indexing XML data. They have defined Minimal Infrequent Structures (MIS) which are structures that exist in the data, not frequent with respect to threshold and wherein all substructures are frequent. Creating an index based on the presence of MIS can be used to locate the selective substructure of a query effectively. During the query evaluation the data mining algorithm based on the concept of MIS efficiently filter the unwanted results.

Chaudhuri et al (2005) designed an efficient search algorithm for avoiding duplicate mapping in the XML data manipulation by considering the dependency between physical and logical design of relational databases. The careful independent analysis of physical and logical design can lead to provide suboptimal performance improvement. The search space is reduced by using workload information.

Yoon (2006) suggested a bitmap indexing scheme for XML document. Access control policies in the form of subject, object and action are transformed into bitmaps in the same manner in which the XML indices are created. The bitmap index along with access control policies are pipelined to the faster access of the XML document in a secured way.

A keyword proximity query that reruns the set of Minimum Connecting Tree (MCT) of the matches to the individual keywords in the query is designed and implemented by Hristidis et al (2006).

The string manipulation problems such as inconsistency between variables, such as misspelling will exist in the XML document. Guha et al (2006) proposed an approach to handle the problem of integration of data from multiple sources. They addressed the problem of integration of XML data sources through correlation realized as join operations. Two XML documents with different structures convey the same information. This is
realized with approximate matching in a structure in addition to the content to perform a join operation. The approximate structure match and content for pairs of documents are analyzed with the well defined notion of distance. The proposed framework model shows the metric properties of notions of distance to join between different XML data sources with the idea of the reference set. Sampling based algorithms are used to identify the reference set. The proposed framework is combined with the well known indexing structures to perform the approximate join operation efficiently.

The XML documents stored in a peer-to-peer network model can be efficiently located by the system implemented by Rao and Moon (2009). The proposed system called psiX runs on top of the existing distributed hashing framework. The structural summary of each XML document is captured and transformed into algebraic signature. The query pattern is also mapped into the signature. Both the document signature and query signature are compared for matching to find out the relevant document. The query pattern is processed as a whole pattern. All the participating peers collectively maintain the distributed hierarchical indexes for the document signatures.

Liu et al (2010) developed a technique to search keywords from XML documents and specify the inference associated with the keywords. The retrieved keywords are clustered based on the semantics inference.

Keyword search is an effective paradigm for information discovery and has been introduced by Liu et al (2011) to query XML documents. In this approach they have addressed the problem of returning clustered results for keyword search on XML documents. They proposed a novel semantics for answers to an XML keyword query. The core of the semantics is the conceptually related relationship between keyword matches, which is based on the conceptual relationship between nodes in XML trees. Then they proposed a new clustering methodology for XML search results, which
clusters results according to the way they match the given query. Two approaches to implement the methodology are presented. The first approach is a conventional one, which does clustering after search results are retrieved; the second one clusters search results actively, which has the characteristics of clustering on the fly. The generated clusters are then organized into a cluster hierarchy with different granularities to enable users locate the results of interest easily and precisely. Experimental results proved the proposed semantics as well as the efficiency of the proposed methods.

A labeling scheme (Hamdi et al 2012) proposed for updating XML siblings dynamically. It is used to identify the relationship existing between nodes in XML tree based on the relationship that only a few nodes require node re-labeled during update operation.

1.3.3 Existing Works on Version Management

The management of multiple versions of XML documents finds important applications and poses interesting technical challenges for database researchers. Research on managing multi version XML document seeks to provide efficient and robust techniques for (i) storing and retrieving, (ii) exchanging, and (iii) querying such documents. Traditional version control methods such as Revision Control System (RCS) (Tichy 1985) and Source Code Control System (SCCS) (Rochkind 1975) fall short of satisfying these three requirements. The SCCS is a software tool designed to support programming projects to control the source code. It helps in storing, updating, and retrieving all versions of modules, for controlling updating privileges, for identifying load modules by version number, and for recording each change made in the software by whom, when and where it was made and why.

Many traditional document versioning systems, such as RCS (Tichy 1985) are editing based. They use edit scripts to represent document
changes and to reconstruct different versions incrementally. For instance, RCS (Tichy 1985) stores the most current version intact while previous versions are stored as reverse editing scripts. These scripts describe how to go backward in the document’s development history. Instead of appending version differences at the end like RCS (Tichy 1985), SCCS (Rochkind 1975) interleaves editing operations among the original document / source code and associate a pair of timestamps (version ids) with each document segment specifying the lifespan of that segment. Versions are retrieved from an SCCS (Rochkind 1975) file via scanning through the file and retrieving valid segments based on their timestamps.

In the database community, there is much interest in XML as a general vehicle for data definition, and in powerful query languages for XML documents. In this context, XML versioning becomes yet another instance of the various issues pertaining to the transaction-time databases (Ozsoyoglu and Snodgrass 1995). A temporal database contains time-varying data. In real-time, database transactions have deadlines or timing constraints. In this context Ozsoyoglu and Snodgrass (1995) reviewed the substantial research in these two previously separate areas, characterized into the time domain and then investigated temporal and real-time data models. The temporal and real-time query languages are evaluated along with several dimensions and examined temporal and real-time DBMS implementation. Finally, the summary of major research accomplishments to date and list several unanswered research questions are reported.

Chawathe et al (1996) designed a model to manage the versions of hierarchically structured documents. They defined the hierarchical change detection problem as the problem of finding a "minimum-cost edit script" that transforms one data tree to another, and present efficient algorithms for computing such an edit script.
Version management techniques proposed for object-oriented databases and semistructured information (Chawathe et al 1996) assume that the order between objects is not significant, while this is essential for reconstructing an XML document. Ramakrishnan and Janakiram (1996) suggested a model for managing the versions of objects used in design databases such as Computer Aided Design (CAD) / Computer Aided Manufacturing (CAM) etc. Hatz and Change (1987) proposed a framework model for managing the changes in CAD/CAM databases.

The detection and representation of changes in data is very much crucial for active databases, data warehousing, view maintenance, and version and configuration management. The existing schemes are used to detect the changes in flat file data and relational data only. Chawathe et al (1996) suggested a scheme to detect and represent the changes in the hierarchy structured data. It is based on computing the minimum-cost edit script that transforms one data tree to another tree. An efficient algorithm for finding minimum-cost edit script is presented. The performance of the proposed scheme is better than the existing schemes used to detect the changes in the document.

Chien et al (2000) proposed a system for managing the versions of XML documents with efficient storage and fast retrieval. The traditional technique RCS is used to model the documents as a sequence of lines of texts and small edit scripts to denote the version differences. The logical structure of the original document is not preserved and inefficient. In this system, the structure of the document is preserved and the sub-objects are time stamped in hierarchical order for reconstructing the past and current versions efficiently. This approach is called Usefulness Based Copy Control (UBCC). It is used to retrieve the partial documents and as well as maintaining the multiple concurrent versions.
The performance of UBCC (Chien et al 2000) system is analyzed and compared with editing based system by them in 2001. The algorithm based on the page usefulness management enhances the performance of UBCC.

The reference based versioning with an edit script model developed by Chien et al (2001) includes temporal clustering at physical level. It is used to preserve the basic order of the document by identifying the objects shared with earlier versions. This model reduces the average version retrieval cost with the additional overhead of storage space.

Chien et al (2001) suggested a system to manage the multiple versions of documents based on object referencing. The reference based versioning scheme captures the evolving rich logical structure of the document. This approach provides a better support for queries and manages the storage level and transport level representation of multiversion of documents. An efficient algorithm is also presented for supporting projection, selection, and document evolution queries.

Marian et al (2001) presents a change centric method to manage versions in a Web Warehouse of XML data. The first step is a sequence of snapshots of XML documents collected from the web. The diff algorithm is executed to compute the changes between two immediate versions. The changes in the form of deltas are represented as sequences with persistent identifier. Basic mathematical foundations for physical storage policies and logical representations are presented.

In recent days, mobile handheld devices are used for business transactions. Data stored on the server are retrieved by multiple clients and updated. In order to provide consistency, an efficient method to synchronize the XML data on multiple mobile devices was proposed by Lam et al (2002).
Each mobile device receives the copy of data from the server based on the given path expression. In order to avoid the unnecessary synchronization overlapping and disjoint copies of data are eliminated. All handheld devices which are affected by the update operations are identified and synchronized.

Wong and Lam (2002) developed a system for version management of XML data. With the support of this system, user can query previous versions, query updates in documents and retrieve a specific document version efficiently.

A research work related to detecting the changes in XML was carried out by Cobena et al (2002). The algorithm implemented is efficient in detecting changes in the XML document in terms of speed and memory space but it loses little quality. They presented the test report of the proposed algorithm for XML pages found on the Web. Wang et al (2003) developed an algorithm called X-diff to effectively detect the changes between XML documents.

In the reverse engineering and maintainability of legacy systems, it is essential to provide traceability link recovery between source code and documentation. The recovered link must be analyzed for the conformance over time because of system evolution. Due to the evolution, the validity and the conformance of the link also has changed. Maletic et al (2003) presents a formal framework model to combine the link recovery with conformance analysis that is supported by the formal hypertext model. The hypertext model supports complex linking structure and versioning of individual links. The results of the traceability link is preserved and maintained by this model for over a time.

Wong and Lam (2003) suggested a model for the reconstruction of XML document versions. The changes in the form of deltas can be used to
construct both forward and backward versions. Both space and time complexity is reduced by mapping between forward and backward deltas.

Vagena et al (2004) developed a system to manage the versions of XML document based on path stack and optimal pattern matching algorithm. The path expression queries can be evaluated over XML documents with branched versions. The storage schemes efficiently support the maintenance of all branched versions and specify the changes required.

A concurrency control transaction protocol for XML data is proposed by Dekeyser and Hidders (2004) based on conflict scheduling. This scheduling algorithm based on path lock and improved model of commit scheduler called conflict scheduler. It supports serializability.

The scheme for management of version control of office documents proposed by Ronnau et al (2005) is based on structured version control. The collaborative work related office documents encounter an obstacle when the version control system store the documents as binary objects. The Application Programming Interface (API) model was implemented in open office versioning can be used as a replacement for line based or binary diffs, which are currently used.

Chen et al (2005) developed a model for internet based distributed computer aided design systems. The major issues addressed by this model are platform dependency and adaptable structure, distributed design control and cooperation, design process representation, collaborative design synchronization and visualization design. This model takes the collaborative design packages added to the applications running on different operating systems. The main characteristics of collaborative computer aided design are horizontal or hierarchical in design activity. In horizontal design activity, the design team from the same discipline completes the design task concurrently.
The hierarchical collaboration is used to establish a strong effective communication path between the higher and lower level of design teams. The system proposed by Chen et al (2005) addressed the following issues prevailed in the collaborative design.

- Synchronization
- Concurrence
- Optimization
- Verification
- Notification
- Manipulation
- Modification

The collaborative design system used in engineering must fulfill these requirements.

- Viewing functions with modeling operations should be provided for the creation of product models. A group of people working in an environment separated from the location can be able to update the artifact data through online.

- The information transferred between group of users should be reduced to the maximum extent possible, to use less bandwidth of the internet connection.

- The design procedure along with a familiar system for the user must be adopted.

- Environment should be heterogeneous.
To meet the requirements, Li et al (2007) developed a system model for real-time collaborative design platform of heterogeneous computer aided systems. This model is command based and so the amount of information exchange is greatly reduced. The design of an artifact is constructed and modified synchronously from the network of workstations in the collaborative environment. In order to support cross platform the commands used in the individual systems are translated into neutral modeling commands and neutral modeling commands are translated into system modeling commands for real time collaborative design. Neutral modeling commands provide platform independence.

In this collaborative design model neutral modeling commands are only communicated to another system for real time synchronization. Whenever the system receives neutral modeling commands, it immediately transforms into system modeling operation. This model is based on the client server architecture model with a central server connected with several clients to form an integrated system. On every client, system is equipped with a manager add-on, which translates the system modeling commands to neutral modeling commands. This neutral modeling commands are sent to the server and then passed to all the other client systems. The major contributions of this system are

- New method for creating an online collaborative design platform is presented.

- The collaborative design platform is based on the integrated approach with heterogeneous CAD systems. Multiple users at different workstation can manipulate the models by their CAD systems during the design process.
• Visualization with real-time manipulation of design of model is supported by the system, which fulfills the requirements of the industrial users.

• Several users can simultaneously collaborate in a real-time manner to create and manipulate models. Since the neutral modeling commands are exchanged between the systems, it requires only a limited bandwidth for interaction.

The engineering design process includes a team of designers to share the design details with others for a collaborative design. The complexity of the artifacts need the distributed, collaborative, and integrated design to meet its requirements. Major issues related to collaborative design are due to: a group of people, different disciplines, of different enterprises, at different locations, having to work together on one artifact. Complex artifact consists of a set of simple subcomponents. The design of artifact requires an expertise, knowledge, technology and tools. Resources required for the collaborative design is distributed.

Autexier and Muller (2010) implemented a framework model in the tool GmoC. It is used to semantically annotate the collection of documents and assess the effect of changes made in different document collections. Large collection documents are generated and modified every day. The documents contain rare isolated artifacts but are related to other documents. This relationship needs some form of adaptations with another document, when one document is changed. The framework is implemented with the existing document types, declarative specifications of semantic annotation and propagation rules inside and across documents of different category. It captures the change impact analysis of heterogeneous collections of documents.
1.3.4 Literature Related to XML Security

The flexible format of XML allows applications to exchange data across the internet. For use of XML in e-commerce and online application, it is necessary to protect the data from unauthorized users. It is needed to protect the integrity of information exchanged between applications over the internet. The security mechanism is essential for XML data to provide confidentiality and authentication. The World Wide Web consortium addresses these issues through XML encryption, XML digital signature and key management. In this section the related literature of XML security are considered.

Real world use of XML is widespread due to the fact that XML is the de facto standard for data exchange in information technology not only in the well-known areas, such as E-Commerce, Grid Computing, Agent Systems, etc., but also in a lot of major industry sectors, e.g. the automotive industry, which focus on XML based description standards for data interchange. Its use in internet e-commerce applications demands the essential component of all electronic security systems such as confidentiality, authentication and integrity. Confidentiality plays the most important role for the exchange of information between two business partners. The flexibility, scalability and confidentiality can be provided easily, but only at the expense of performance. Public-key cryptography (Rivest et al 1978) provides a mechanism to fulfill all three requirements.

An encryption method (Pollmann 2002) for XML which is capable of encrypting single XML Information Set items. This method is able to hide the size and the existence of encrypted contents. It prevents the traffic analysis. The W3C has launched the XML Encryption working group in the year 2001. The role and responsibility of this forum is to define how to encrypt the portions of XML documents. It describes an encryption system
which allows to have these 'deep children' in plain text while having the ancestors encrypted, i.e. bringing the property from XML Access Control to XML Encryption. The portion must always be a subtree or a consecutive sequence of subtrees. The XML Access Control allows more granular restrictions on what portions on an XML document a client is allowed to see. XML Access Control can remove an ancestor node from a document while leaving a descendant node in the document. The implemented system which allows to have these 'deep children' in plain text while having the ancestors encrypted, i.e. bringing the property from XML Access Control to XML Encryption.

The large quantity of information is presented in XML format on the Web, causes there are increasing demands for XML security. Research on XML security has been focused on the security of data communication using digital signatures or encryption technologies. As XML is also used for a data representation of data storage, XML security comes to involve not only communication security but also managerial security. Managerial security is guaranteed through access control, but existing XML access control models consider only read queries. These models may make some problems when unauthorized users try to change XML documents or their structure. Therefore the access control of update queries must be executed correctly and efficiently as well as read queries.

To protect the XML data description, access control model (Lim et al 2003) was implemented that supports read and update operations. A new action types are defined to systematically manage complex information access rights and manage update queries in an efficient manner. It focuses on XML access control model and present a technique that supports not only read operations but also update operations. A new action types are defined to systematically manage complex information to access right and to process
various update queries in an efficient manner. Using these action types, the system can save memory and other system resources that are used in DOM-based DTD verification process, and shortens the overall steps of access control by filtering unnecessary queries out at the earliest stage. Although for read queries, the implemented access control model introduces a minor overhead in determining action types and for update queries it shows better performance compared to the existing access control models.

The security mechanisms used in an application should not significantly reduce the system's performance but it should greatly enhance its application scope. Peer-to-peer security mechanisms such as SSL and TLS are suitable for ensuring data integrity, confidentiality and authenticity but not non-repudiation. Ensuring non-reputability of multi-signed digital documents is a primary requirement for the university's examination mark system. A general XML schema for the signed document and its elements to ensure integrity, confidentiality, authenticity as well as non-repudiation has been implemented (Barhoom and Zhang 2004). This system has been used as an Examination Management System at University Level Examination Management.

An operational model for XML document security (Hwang and Chang 2004) is presented. For the given XML document \( X \), the operational model defines the process of encrypting data and embedding digital signatures which sign the data in \( X \). The secured XML document \( X_s \) includes encrypted and unencrypted data of \( X \), and embedded digital signatures. The operational model also defines the processes of decrypting \( X_s \) and verifying the digital signatures embedded in \( X_s \). It offers a security mechanism which integrates element-wise encryption and temporal-based element-wise digital signatures. This operational model provides an element-wise encryption that
is more general than previous forms of XML security, by including element encryption, content encryption, and two types of attribute encryption

Based on the generalized operational model, a new language called Document Security Language (DSL) is defined. The syntax of the encrypted document and the equivalent transformation language are presented. To facilitate the automation DSL includes the definition of the standard DSL algorithm downloading and linking protocol that fulfills automatic algorithm download linking requirements. This model based on DSL tool is configurable. Two different implementations further demonstrate its practicability: while one uses the Java programming language to implement the securing tool, the other uses the extension mechanism of XSLT 1.0 to implement the encryption and decryption transforms. The two implementations are available free on the Internet. Experimental results obtained when using this securing tool demonstrate the automation, efficiency, and practicability of the defined operational model. In addition to that, a DSL editor with a friendly graphic user interface to make it easier for users to generate DSL documents is also implemented

A system (Zhang et al 2004) for providing security for web services based on XML security key technologies has been defined. In this system web service is integrated with some mature security architectures such as PKI. The characteristics of web service layer security model are described. In this paper public key encryption of XML data under different scenario is analyzed and the performance results are presented.

Web Services consist of a process and set of protocols for determining and interconnecting to the software exposed as services over the internet. Web services are built on Simple Object Access Protocols (SOAP). It provides the mechanism to get / access data from point A to Point B. Web service facilitates the software interaction. The technical infrastructure of web
services support the component from different vendors, different platforms will be integrated to form a complete business solution. The key technologies used for web services are exhibited (Beznosov et al 2005). The main building blocks of web services are SOAP, Universal Description Discovery and Integration and Web Services Description Language. The main focus is on technologies for protecting the data.

Signatures of SOAP/XML messages are necessary to enforce the security. An approach is specified (Chen et al 2005) to accelerate the process of validating signatures of Web service calls. A system called Quick XML parser (QXP) has been implemented. The QXP is motivated by Xerces, but QXP outperforms Xerces ranging from 7 to 21 times speedup in the two benchmark programs.

A method for Partial RDF Encryption (PRE) (Giereth 2005) is developed in which sensitive data in an RDF-graph is encrypted for a set of recipients while all non-sensitive data remain publicly readable. The result is an RDF-compliant self-describing graph containing encrypted data, encryption metadata, and plaintext data. For the representation of encrypted data and encryption metadata, the XML-Encryption and XML-Signature recommendations are used. The implemented method allows for fine-grained encryption of arbitrary subjects, predicates, objects and sub graphs of an RDF-graph. An XML vocabulary for specifying encryption policies is introduced.

A new design key management protocols and services with unified single interface, which extend the scheme and the services of XKMS, for various kinds of security information such as, private key, secret key, biometric information, security tokens for Web service was specified (Kim and Moon 2005). This model modifies and extends the protocols and services such as key registration, reissue, revocation, recovery, location service and
validation for only public key pair to provide a way to manage other security information including public key pair.

One of the most important cryptographic primitives is a digital signature. It supports the data integrity, message authentication and non-repudiation, which are essential attributes in security critical services, such as e-commerce, voting or health care or credit card payment. The previous data formats for digital signatures concentrated on signing the entire document, the XML signature standard is feasible to secure complex workflows on a document with multiple signatures. A research work focused a new model (Kubbilun et al 2005) with multi signed XML documents that is used particularly in the field of presentation. The use of XML Signature may result in signed documents remaining vulnerable to undetected modification by an intruder. The signed documents are realizable in standard web browsers.

An intruder may be capable of modifying valid messages in order to gain unauthorized access to the protected resources. The possible general vulnerability and several related exploits are presented (McIntosh and Austel 2005), and appropriate countermeasures are defined. It gives a wide range of guidelines to prevent all possible attacks.

A key management architecture model (Park et al 2005) based on XML standards for securing access to location based information and services was implemented. The main property of this approach is that it is built on a number of XML-based standards. The critical application of wireless internet, the Location-Based Services (LBS) provides location based services with the support of middleware servers. In this category of services, security is very important to protect the stored location information in mobile devices and provide support for services based on the location information. Safe security must be provided for the collection of location information, user management
that includes authentication and information security and the access control of a large amount of information stored in the database.

The basic features of XML security plug-in and integration with CrypTools are presented (Schadow 2005). The XML Security Plug-In is a freely available plug-in for the Eclipse platform and provides versatile XML security features in an educational software like environment. End users can experiment with XML security and inform themselves about the corresponding W3C recommendations. The objective of plug-in is to increase the XML security functions and specifically to encourage the user to sign and encrypt their vital information and critical messages.

SOAP applications adopt the XML signature specification that support rich and flexible message signature model for XML documents to support message-level security (Wei et al 2005). The design of XML signature raises a number of complex processing steps such as canonicalization and XPath filtering that lead to performance degradation and scalability issues. The focus of this work is to evaluate the performance of validating large signed XML messages will be sent by scientific applications with the help of Grid Web Services. The design and implementation of the GHPX/SSSV system has been done for the streaming validation of SOAP signature validation. This system consists of a streaming canonicalization and optimized SOAP signature validation. The empirical performance study of the characteristics of the streaming validation is done and it shows that it provides high performance and also memory efficient.

A server security certificate management system (Kim et al 2006) applying the mechanism of public-key infrastructure and XML Security technology specification to secure the information and resources open in the network. This system that permits the access only to the authorized users at a request of the status of the certificate through certifying server after issuing
and requesting wire, wireless certificate online and registering on the XML certificate managing server.

The most important technology in the electronic commerce based on the Internet is to guarantee the security of trading information exchange. An XML security model for web services (Cho 2006) based electronic commerce system to guarantee the secure exchange of trading information is defined. To accomplish the security of XML, the differences of XML signature, XML encryption and XML key management scheme with respect to the conventional system should be provided. The new architecture is specified based on the unique characteristics of XML. Especially the method to integrate the process management system need to the electronic commerce is proposed.

Multi-agent systems were used in proprietary environments. These systems have been used broadly in open distributed networks, such as e-commerce applications for the Internet. Multi-agent systems should have security mechanisms, e.g. confidentiality and integrity. A model for secure communication (Oliveira et al 2006) for agents based on RDF and the XML security specifications is presented. Agents may use Foundation for Intelligent Physical Agents (FIPA) standard called Resource Description Framework (RDF), which is a message content standard in XML language. Using this standard, agents can communicate exchanging XML messages, but these messages are not secure.

The open security architecture (Park et al 2006) for Location based Services (LBS) platform is implemented to provide interoperability among the wireless networks and various location-based application services and the functional security requirements for the LBS platform. The objective of this architecture is to investigate how well the most limited wireless devices can make use of LBS security services. Security acceleration methods for high
speed open LBS using XML signcryption mechanism is also specified. This secure LBS security protocol allows the client to assign the mobile certificate handling in the server and to activate the central administration of privacy policies. The software based signature acceleration is provided by signcryption technique.

The most XML encryption techniques consider the confidential parts of an XML document as a whole block of text and encryption technique is applied directly on the text. Due to this, query in the encrypted part cannot be efficiently handled. To solve this problem a novel approach called XQEnc has been implemented (Yang et al 2006) for XML encryption for better efficiency and robust security. It is based on the concept of vectorization and skeleton compression. Vectorization, which is a generalization of columns of a relational table that utilizes the path pattern of XML tree to label the data values. The compression technique skeleton modifies the redundant path into a multiplicity attribute values.

The encryption and signature standards proposed by W3C specifying the format for encrypted XML documents is important advances towards XML security. Despite the success of these XML security works, it can be considered them to be insufficient from the viewpoint of software engineering. It is necessary to design an appropriate API for the securing system of subtree encryption for XML documents was implemented (Chang and Hwang 2007). The goal is to increase productivity and reduce the cost of maintaining this kind of software, for which a Document Security Language (DSL) was linked to the API. The implementation details of the DSL API, and the experimental results demonstrate its practicality is reported.

In order to improve the performance, the system model (Ammari et al 2011) is developed based on the importance of the message and select the part of the message to be encrypted. Flexibility, friendliness, and
adaptability have been key components to use XML to exchange information across different networks providing the needed common syntax for various messaging systems. This research work presents a novel approach to secure XML messages being used in various systems with efficiency providing high security measures and high performance. This system model is based on two major modules, the first to classify XML messages and define which parts of the messages to be secured assigning an importance level for each tag presented in XML message and then using XML encryption standard specified W3C to perform a partial encryption on selected parts defined in classification stage. It aims to improve both the performance of XML encryption processes and bulk message handling to achieve data cleansing efficiency.

A file based encryption has been defined (Chong and Siong 2011) for securing the XML data flow and exchange between the applications using public key infrastructure technology supported by the W3C working draft for encryption. The most popular technology about the XML is the feature of structuring data and the XML based encryption in a natural way to handle complex requirement for securing the XML data flow and exchange between applications. This research work specifies the implementation of XML encryption utilizing The Public Key Infrastructure (PKI) technology, compliance with W3C's working draft for XML encryption is implemented for XML security.

A new public-key algorithm (Nithin and Bongale 2012) based on multi prime RSA technique was implemented. There are various cryptographic algorithms defined and implemented to encrypt XML documents to address the issue of security. This literature focus on a new public key cryptographic algorithm called XML Batch Multi-Prime RSA (XBMRSA) to encrypt sections of XML document based on Multi-Prime
RSA technique. A complete comparative study of standard RSA and XBMRSA is carried out by considering parameters like time required to encrypt and decrypt the XML file and size of the encrypted file. The implemented system XBMRSA has provided better results as it takes less time to encrypt and decrypt the XML data.

1.4 OBJECTIVES

The design of indexing and clustering techniques for databases is a challenging task for researchers and huge research efforts were done in the recent years led to the variety of access methods and storage organizations. This research work mainly achieves the following objectives.

- Designing of a new clustering model for XML document and its implementation
- The performance improvement of clustering, indexing and also the combined approach is analyzed.
- Content based indexing model is proposed for retrieval of XML data from flat files.
- The performance of the indexing model is compared with the existing indexing techniques in commercial databases.
- New version management model called object versioning is proposed to manage the versions of XML data stored in files and exchange between data in the XML format.
- The performance of public-key encryption used for XML data exchange is analyzed.
1.5 THESIS ORGANIZATION

The remainder of this thesis is organized as follows:

Chapter 2 presents the clustering method proposed in this thesis with the results. Chapter 3 describes the indexing model proposed in this thesis along with algorithms proposed and provides the results obtained in this research work. Chapter 4 details the object versioning model used for managing the versions of XML data. Chapter 5 gives the performance analysis of public key encryption algorithm used in XML data security requirements. Conclusion of this work and some possible future enhancements are given in the chapter 6.