International Journal of Academic Research in Computer Engineering
Print ISSN: 2476-7638 and Online ISSN: 2538-2411
Vol. 1, No. 2, Pages. 05-15, November 2016

	International Journal of Academic Research in Computer Engineering Volume 1, Number 2, Pages 05-15, November 2016 www.ijarce.org	IJARCE

A New Indexing Method for Secure XML Data Broadcast in Mobile Wireless Networks

Madeh Shokri[1*] and Mahvash Shokri[2]

1. Department of Computer Engineering and Information Technology, Iranians University: An E-Institute of Higher Education, Tehran, Iran.
2. Kurdistan Region Electricity Distribution Company, Kurdistan, Iran.

Receive Date 2016.09.19; Accepted Date: 2016.11.10, Published Date: 2016.11.15

*Corresponding Author: M.Shokri (madeh.shokri@gmail.com)

Abstract

With the rapidly increasing popularity of XML, an effective method to transmit XML data over wireless broadcasting environments is urgently required. Secure broadcasting of XML data is becoming an essential requirement for many applications in mobile wireless networks. Several indexing methods have been proposed to reduce the tuning time in processing the XML queries over the wireless XML stream. Tuning time is the sum of period of times which a mobile client stays in active mode in order to retrieve the required data over the wireless stream. Therefore, it is frequently used to estimate the energy consumption of a mobile client. The problem of existing indexing methods is that they cannot directly be applied to an encrypted XML stream since mobile clients can only access the authorized parts of the XML data in an encrypted XML stream. In this paper, we define a new structure for XML stream which supports data confidentiality of XML data over the wireless broadcast channel. We also define an access mechanism for our proposed structure to efficiently process XML queries over the encrypted XML stream. The experimental results demonstrate that the use of our proposed structure and access mechanism for XML data broadcast efficiently disseminates XML data in mobile wireless networks.

Keywords: Indexing, Mobile wireless broadcast, Data confidentiality, Access control, XML streaming.

1. Introduction

As extensible markup language (XML) [1] is emerging as a standard for data dissemination over the Internet, the use of XML for data broadcasting in mobile wireless networks is rapidly increasing. Recently, many applications are using XML for data broadcasting in mobile wireless environments such as traffic and travel information systems and weather information systems [2]. In general, a mobile device can stay in two modes: the active mode and the doze mode. The active mode is when the mobile device receives and processes the data from the relative channel; and the doze mode is when the mobile device does not react to the received data [3, 4].

There are two metrics utilized in order to measure the efficiency of data broadcast in a wireless broadcast channel: the access time and the tuning time [5, 6]. The tuning time is equal to sum of the time periods that a mobile device has spent to receive its required data. Furthermore, the tuning time is a metric for measuring the energy usage. The access time means the spent time from the moment of sending the request by the mobile client to the moment of receiving the complete required data from the broadcast channel, which is presented as the efficiency metric.

Regarding the definition of the access time, it is clear that the existence of irrelevant data in a broadcast channel increases the access time in mobile devices and thereupon, decreases the efficiency of data broadcast. For this reason, a scheduling must be designed in which the time and the content of broadcasted data is specified.

Typically, there are two modes of XML data broadcasting: Push-Based Mode and On-Demand Mode. In the push-based XML data broadcasting mode [2, 7-10], XML data are periodically broadcasted on the downlink 2 channel and mobile clients listen to the broadcast channel and retrieve their desired XML data. In the on- demand XML data broadcasting mode [11-17], mobile clients send their XML queries to the server through the uplink channel and the server considers all the pending XML queries and decides the content and order of XML data for the next broadcast cycle. It should be noted that this paper proposes a new structure and access mechanism for secure XML data broadcast in the push based mode.

Nowadays, the XML data broadcast in a secure way in mobile wireless networks is an essential requirement for many of the applications that are used in this environment [20, 21]. Several approaches have been proposed to broadcast the XML data in the Internet such as approaches proposed in [23-29] but these approaches cannot be applied to the mobile wireless networks for the following reasons:

- The conventional approaches to XML data broadcast are used in networks with high bandwidth and data processing capacity while in mobile wireless networks, mobile clients use battery powered hand-held devices with low data processing capacity.
- The main concerns of the conventional approaches in XML data broadcast in the Internet are efficiency and scalability while the main concerns of broadcasting the XML data in the mobile wireless networks are access efficiency and energy conservation of mobile devices.

Several indexing methods have been proposed to selectively access to broadcasted XML data in mobile wireless networks [2, 7-10]. The main purpose of the indexing methods is selectively access to desired XML data among the whole of broadcasted XML data in mobile wireless networks with the aim of reduction in the energy use of mobile devices for receiving the desired XML data over the broadcast channel. On the other hand, for XML data streaming in wireless networks, the XML tree nodes must be placed in sequential order on the data stream. This sequential access causes delay in receiving the data over the broadcast channel. Indeed, using an index it is possible to reach exactly the desired XML data without receiving all of the XML data in the XML stream. In the other words, instead of receiving entire the XML data, only the part which is needed will be found by the index and retrieved. As a result, the index can save energy consumption to receive data on the mobile devices. The problem of delay in receiving the XML data in a mobile wireless network is due to the sequential access to data on the broadcast channel.

The first proposed structure for XML data broadcast in mobile wireless networks is the S-Node structure [7]. For the S-Node structure, three different indexing methods called OSA, TSA, and SPA are proposed which can be used in processing the simple path XML queries at mobile clients. A new indexing method was proposed in [18] using the structural summary of the XML tree. The proposed indexing method in [18] is more efficient than the indexing methods proposed in [7] because the path summary structure used in [18] reduces the tuning time and the access time much more than the S-Node structure in [7]. However, the proposed indexing method in [18] is able to only process the simple path XML queries.

To solve the delayed query processing problem, a new structure is proposed in [2]. This structure which is called DIX-node includes the XML data and various XML indices. Since the DIX-node contains the location path information, mobile clients can process the XML queries without the necessity of wait for the arrival of the next broadcast DIX-node. Also, by exploiting the clone node link and the foreign node link as the indices, they are possible to skip from the irrelevant nodes over the XML stream. However, the proposed indexing method can only process the simple path XML queries.

The processing of twig pattern XML queries has been investigated in [19] which proposed the G-Node structure. The G-Node is a data structure that contains the path information, the group explanations, all the attribute values, and a list containing all the text content of the element. In [19], the Lineage Encoding has been proposed as a new manner of encoding which is utilized in the processing of twig pattern XML queries. The Lineage Encoding includes two encodings: the vertical or the Lineage Code (V) and the horizontal or the Lineage Code (H). Using the Lineage Encoding, mobile clients have ability to find the results of twig pattern XML queries with performing the bitwise operation on the LC in the relevant G-Nodes.

In [10], a novel structure is proposed for streaming the XML data called PS+Pre/Post by integrating the path summary technique [30] and the pre/post labeling scheme [31]. The proposed XML stream structure exploits the benefits of the path summary technique and the pre/post labeling scheme to efficiently process different types of XML queries over the XML stream. The experimental results show that the PS+Pre/Post structure is capable of processing all types of XML queries; furthermore, has a better performance than the other XML data broadcast methods.

An approach to disseminate the encrypted XML data has been proposed in [24] that has the capability of hiding the size and the content of XML data. The main idea of this approach is that all of the XML nodes which need to be encrypted are encrypted separately and then they are put in a pool of encrypted XML nodes. This pool can be stated in the same document near the unencrypted XML nodes or be stated in the other documents. The encrypted XML nodes find the way to return to their suitable place magically at the time of decryption. In the proposed approach, every XML node is encrypted via a unique key. In order to transfer these encrypted XML nodes to the receiver, the keys are placed in a set of node keys and this set is encrypted again by the key of the receiver [25].

The concept of query-aware decryption is presented in [26]. The query-aware decryption lets to decrypt only some parts of the XML data which are of the query results. For this purpose, it is necessary to broadcast the encrypted index besides the encrypted XML data. After decrypting the index, the place of the query results in the encrypted XML data will be found. Therefore, it is no needed to decrypt all the encrypted XML data. It should be noted that the cost of decrypting the index is negligible compared to decrypting the unnecessary XML data since the size of index is much smaller than the size of encrypted XML data.

A specific broadcast approach for XML data is proposed in [28] called XFlat. The XFlat in comparison with other approaches focuses more on the efficiency of the query on the XML data broadcast and simultaneously protects the important XML data via the encryption methods.

Although these approaches securely disseminate XML data in the Internet, they cannot be used for XML data dissemination in a wireless broadcast channel since the main concerns considered in these approaches are efficiency and scalability while the main concerns in the mobile wireless networks are access efficiency and energy conservation of mobile devices. In addition, the data access over wireless broadcast channel is sequential.

A structure called SecNode has been proposed for secure XML data broadcast in wireless mobile networks which contain two indices called Min(NCS) and Min(NIS) [20, 21]. These indices have been defined based on a set of access control rules which are applied to the XML document. The Min(NCS) index is used to jump to the next candidate accessible node and the Min(NIS) index is used to jump from the next irrelevant nodes that is accessible with query.

2. The Proposed Structure of XML Stream (Secure G-node)

To publish XML data to securely we have a new structure called Secure G-Node for streaming XML data suggest that, in fact, can tuning time, access time to a minimum and also query processing queries with pattern twig and can also support the concept of data confidentiality and access control on XML data to mobile user's implement.

Generally, an XML document can be modeled by a tree structure. In this tree structure, elements are represented by nodes and Parent–Child (P–C) relationships between the elements are represented by edges. Figure 2 shows the XML tree corresponding to the XML document of Figure 1.

```
<mondial>
<country id ="f0 _162" name="Belgium" capital="f0_1477" population
="10170241">
<name>Belgium</name>
<province id ="f0_17457" name="ntwerp">1610695</province>
<province id ="f0_17462" name="Hainaut">
<located at>West of Belgium</located at>
<city id="f0_2335" country="f0_ 162" province="f0_ 17462"
>Charleroi</city> <population >255634</population >
<city id="f0_2345" country="f0_ 162" province="f0_ 17462" >Mons
</city> <population >188639 </population>
<city id="f0_2345" country="f0_ 162" province="f0_
17462">Liege</city> </province> </country>
<country id ="f0 _174" name="Bulgaria" capital="f0_1487" population
="8612757">BU</country>
<country id ="f0 _208" name="Finland" capital="f0_1507" population
="5105230"> <province id ="f0_33615" name="Aland">
  <city id="f0_35399" country="f0 _208" province="f0
_33615">Mariehamn</city>
<population>315630</population >
</province>
<province id ="f0_336 20" name="Haeme">662000</province>
</country><country id ="f0 _184" name="Czech Republic"
capital="f0_1493" population ="10321120">
<province id ="f0_17473" name="Jihomoravsky">
<located at>East Cost of Czech</located at>
<city id="f0_2394" country="f0_ 184" province="f0_
17473">Zlin</city>
</province > <province id ="f0_17475"
name="Severocesky"></province>
</country>
</mondial>
```

Figure 1. An Example of XML Document

To generate an XML stream in a secure manner, the content of XML data must be transformed from the original XML structure to an appropriate format in which the information of the index for selective access to XML data over the encrypted XML stream exists [20, 21].

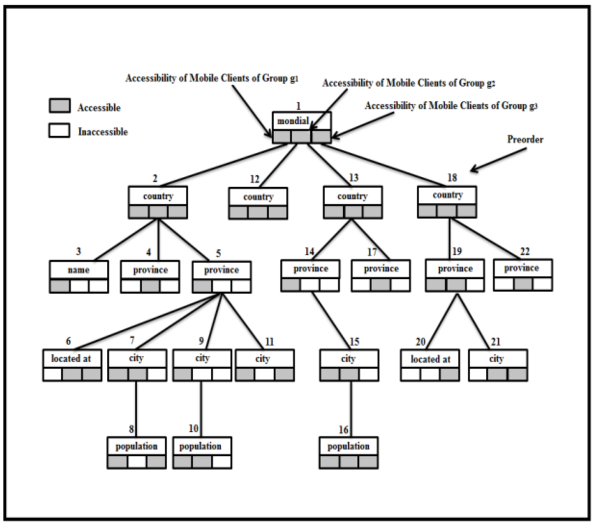

Figure 2. An Example of XML Tree

Let us consider an XML tree as shown in Figure 2. In Figure 2, each XML node is assigned with three rectangles where the left, middle, and right rectangles represent the accessibility of the XML node to mobile clients of groups g1, g2, and g3, respectively. In Figure 2, the gray rectangles represent accessibility of the XML nodes to mobile clients of the different groups while the white rectangles represent non-accessibility of the XML nodes to mobile clients of the different groups. For example, the node "name" is accessible to the mobile clients of group g1 but not to the mobile clients of groups g2 and g3.

2.1. Secure G-node

To process XML documents, we use a technique known as Secure G-node. In fact, a wireless streaming XML data by integrating elements that produce a similar path. This data stream contains a sequence of XML nodes consolidated (Group), is called Secure G-node.

Definition 1. [Secure G-node], The Secure G-node denoted by SG_p=(AC_p,KEY_p,GD_p,AVL_p,TL_p) is a data structure containing information of all the elements e_p whose location path is p, where AC_p is Accessibility Code, Key_p is Encryption key, GD_p is a group descriptor of G_p, AVL_p is a list containing all attribute values of e_p, and TL_p is a list containing all text contents of e_p. Figure 3 shows Secure G-node structure.

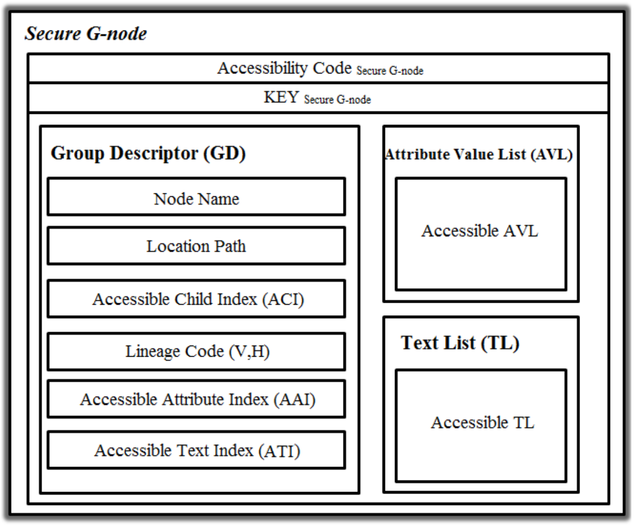

Figure 3. Structure of Secure G-node

When we Secure G-node to the XML tree Figure 2, we can apply a new tree with new nodes (Secure G-node) is created. Figure 4 shows a result of the operation. As used in this figure we summarize attribute and also integrates elements of the path from the root to the same node can be different nodes in the same category come together and Secure G-node that created integrates them.

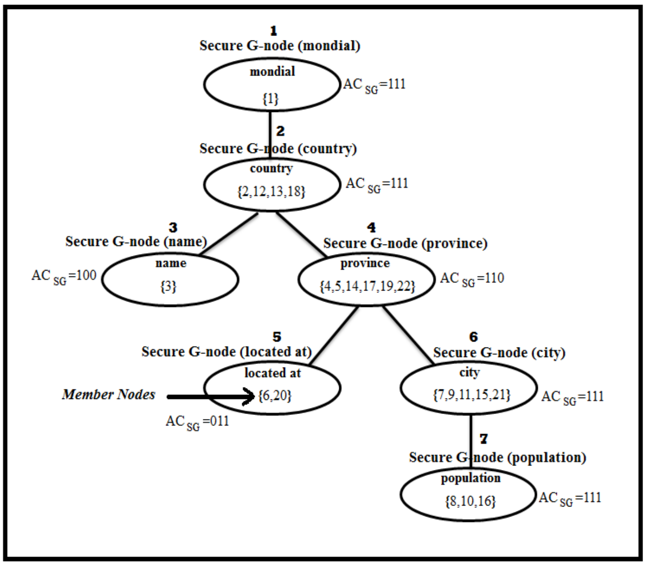

Figure 4. Apply Secure G-node in the Tree XML

The following fields Secure G-node structure, as well as tips and definitions needed to fully explain the charges.

Definition 2. Let U = {u_1, u_2,…, u_n} be the set of users (or mobile clients) and UG = {g_1, g_2,…, g_m} be the set of user groups. We suppose that M: U → UG is a function that maps a user (or mobile client) in U to a user group in UG. The accessibility code of the node e in the XML tree T is defined by AC_e = $a_1a_2…a_m$ where:

1<=i<=m

And $a_i = \begin{Bmatrix} 1, \textit{if the } i^{th} \textit{ user Group in UG} \\ \textit{and } g_i \textit{ has access to the node} \\ 0, \textit{Otherwise} \end{Bmatrix}$

and m is the total number of user groups in UG.

For example, the node 3 (i.e. node ‘‘name) in the XML tree illustrated in Figure2 can only be accessed by mobile clients who are the members of the group g1. Therefore, the accessibility code of the node 3 is 100 ($AC_{\text{node 3}}$ = 100).

Field 1. Accessibility Code $_{\text{Secure G-node}}$ (AC_{SG}): To check the availability of different user groups to different Secure nodes of AC_{SG} used. In fact, by this code, we specify which user groups or Secure nodes to accessible or inaccessible.

The AC_{SG} for both the following procedure to calculate the following code:

- Calculate the accessibility code for all nodes of a member of the Secure G-node concerned.
- Operator OR-Bitwise, to compute the OR- Bitwise all the ACs related to Secure G-node.

For example, in Figure4, AC_{SG} related to Secure G-node $_{\text{city}}$ is calculated as follows:

- AC compute nodes member:

AC_7=110, AC_9=100, AC_{11}=101, AC_{15}=110, AC_{21}=011

- After calculating AC member nodes, OR-Bitwise operation on all AC calculated for the following apply:

AC_{SG} (city) =110 OR 100 OR 101 OR 110 OR 011=111

AC_{SG} of Secure G-node $_{\text{city}}$ is ACS city = 111.

Field 2. Key $_{\text{Secure G-node}}$ (Key_{SG}): The field $Key_{\text{Secure-G-node}}$ in the Secure G-node is the key identifier used for encryption/decryption of the Secure G-node.

Field 3. Group Descriptor (GD): The group descriptor is a collection of indices for selective access of a wireless XML stream.

Field 3.1. Node name: Node name is the tag name of integrated elements.

Field 3.2. Location path: Location path is an XPath expression of integrated elements from the root node to the element node in the document tree.

For example, the Location path of the node “Name” with the preorder 3 (Secure G-node$_{\text{name}}$) in the XML tree illustrated in Figure 4 is the path “/Mondial/Country/Name”.

Field 3.3. Accessible Child Index (ACI): To specify the child Secure G-nodes of this field is kept available for the current Secure G-nodes used. In fact, ACI is a collection of child Secure G-nodes of the current Secure G-node and are available for it.

Field 3.4. Lineage Code (V,H): To support queries involving predicates and twig pattern matching. In the proposed scheme, two kinds of lineage codes, i.e., vertical code denoted by Lineage Code (V) and horizontal code denoted by Lineage Code (H), are used to represent parent-child relationships among XML elements in two G-nodes.

For more information about the lineage code can refer to Article [19].

For example, Figure 5 shows an example of Lineage Codes in Secure G-node $_{\text{country}}$, Secure G-node $_{\text{province}}$, and Secure G-node $_{\text{city}}$. Note that Lineage Code (V) of Secure G-node $_{\text{province}}$ is defined by 1011 since the elements integrated in Secure G-node $_{\text{province}}$ are mapped to only the first, third, and fourth elements in Secure G-node $_{\text{country}}$. Lineage Code (H) of Secure G-node $_{\text{province}}$ is (2, 2, 2), where each value denotes the number of child elements in Secure G-node $_{\text{province}}$ mapped to the same parent element in Secure G-node $_{\text{country}}$ in document order.

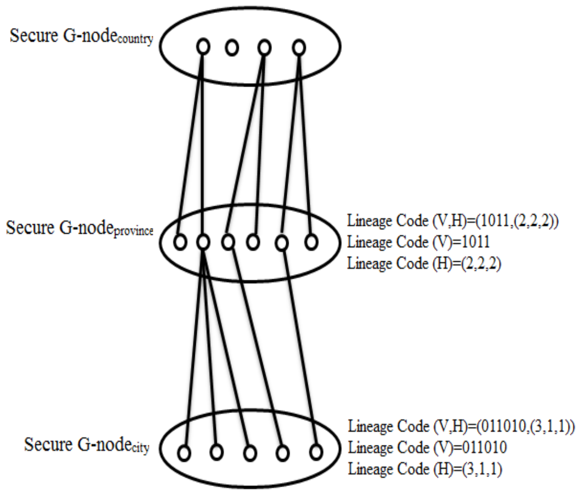

Figure 5. Lineage Codes of Secure G-node country, Secure G-node province, and Secure G-node city

Field 3.5. Accessible Attribute Index (AAI): This field attributes available in a Secure G-node element for the user to specify. In fact, AAI contains the pairs of attribute name and address to the starting position of the values of the attribute that are stored contiguously in Attribute Value List.

Field 3.6. Accessible Text Index (ATI): This field text available in a Secure G-node element for the user to specify. In fact, ATI is an address pointing to the starting position of Text List.

Field 4. Attribute Value List (AVL): AVL store attribute values of the elements represented by the Secure G-node.
Field 5. Text List (TL): TL store text contents of the elements represented by the Secure G-node.
Attribute values and text contents are stored in document order of elements.

3. Query Processing over Wireless XML Stream

In this section, we show that mobile client what it can by using different XPath queries in your desired XML data Using XML wireless streaming of the proposed recovery.

3.1 Simple Path Query Processing

For example, assuming that the mobile client C1 query Q = "/ mondial/country/province [@ name ="Aland "]/city" submit. The mobile client first constructs a query tree, and finds the relevant Secure G-nodes over the wireless XML stream using ACIs. After downloading the group descriptor of Secure G-node province, the mobile client selectively downloads the “name” attribute values pointed by an address contained in AAI because the current query node contains a predicate condition (i.e., name =”Aland”). After downloading of relevant Secure G-nodes, the mobile client C1 computes the result selection bit string by performing the SelectChildren() function (for more information about the SelectChildren() function can refer to Article[19]) from the root node to the leaf node. Note that the selection bit string of Secure G-node province is SB province = 001000 because only the third attribute value in a value list satisfies the predicate and AC province=011010 (i.e., ACI,AAI,ATI province=011010) that according to the AC and SB bit AND-operator actions (i.e., SB province AND AC province = 001000), It is clear that the mobile client C1 has access to attribute related to answer queries (i.e., the value of the attribute to the third element in Secure G-node province). In this example, the mobile client retrieves the fourth element in Secure G-node city because the result selection bit string for the G-node is 00010. Figure. 6 illustrates exploring steps for a simple path query with a predicate over the stream (a) and Lineage Codes computation using a query tree (b).

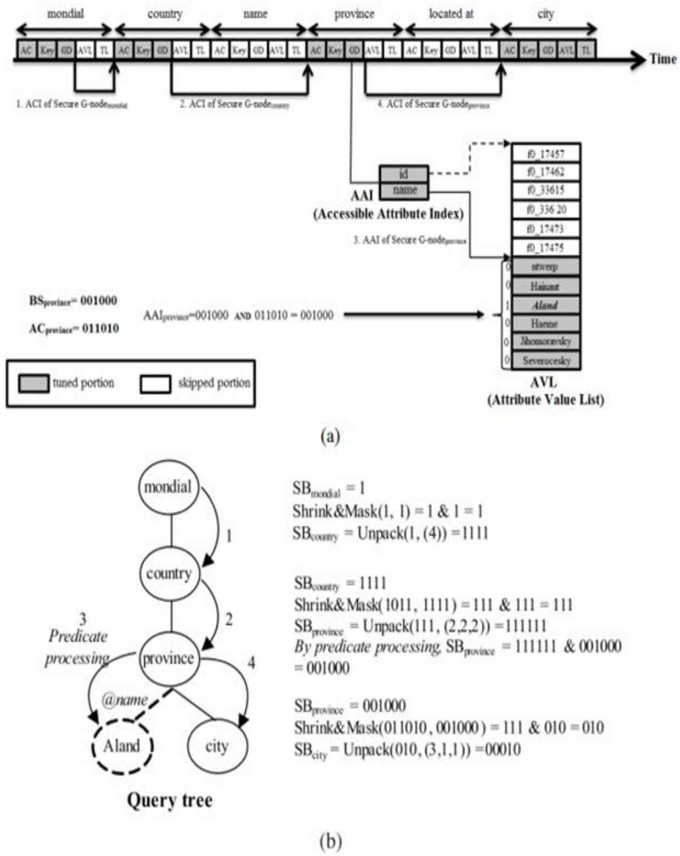

Figure. 6. Example of simple path query processing. (a) Exploring steps over the wireless XML stream (b) The result selection bit string computation on a query tree.

3.2 Twig Pattern Query Processing

Twig pattern query processing consists of three phases: Tree traversal phase, Sub-paths traversal phase, and Main path traversal phase. The main path denotes a path from the root node to a leaf node which represents the target element of the query, while the sub-paths denote branch paths excluding the main path in the query tree.
For example, assuming that the mobile client C1 query Q=”/mondial/country[name]/province/city” submit. In Tree traversal phase, the mobile client downloads group descriptors of five G-nodes (i.e., Secure G-node mondial, Secure G-node country, Secure G-node name, Secure G-node province, and Secure G-node city). In Sub-paths traversal phase, the mobile client C1 computes the selection bit string of the sub-path (1000) using the GetSelectionBitStringOf() function (for more information about the GetSelectionBitStringOf () function can refer to Article[19]). In Main path traversal phase, the mobile client performs the SelectChildren() function from the branching node (i.e., Secure G-node country) to the leaf node (i.e., Secure G-node city). In this example, element nodes 7, 8, and 9 are retrieved because the final selection bit string is 11100 and AC city=11110.

Figure. 7 shows query processing steps for a twig pattern query, "/mondial/country [name]/province/city".

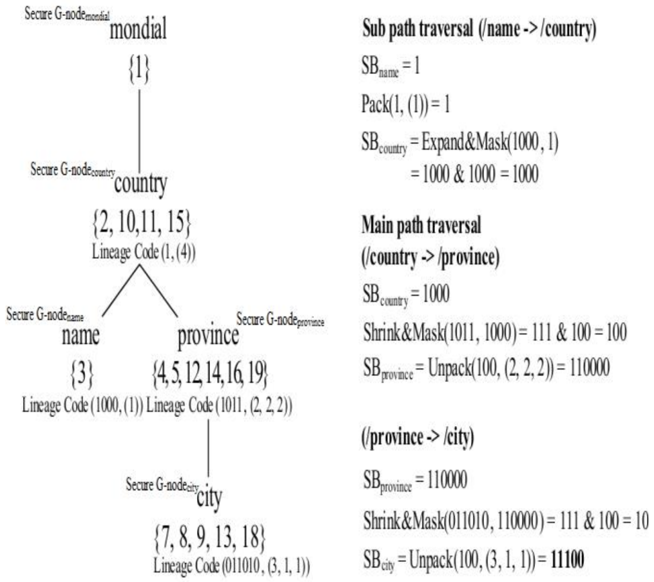

Figure 7. Example of twig pattern query processing

4. Performance Evaluation

The efficiency of our proposed structure to secure broadcast of XML data in mobile wireless networks is investigated in this section. For this purpose, a computer with the processor Intel(R) Core i5-3337U – 180 GHz and 4GB RAM with Windows® 10 Enterprise operating system has been used. All of the codes were implemented in java using NetBeans IDE 8.0.2.

4.1 Experimental Settings

To increase accuracy in computing, the wireless XML data stream has been modeled as a binary file in that the server has written a stream of bytes and the client reads the file and processes its desired queries on it. In this simulation model, it is assumed that the full bandwidth has been used for XML data broadcast. To encrypt and decrypt the XML data, the algorithm 3-DES has been utilized. To measure the performance variation based on the types of XML data sets, several real and syntactic data sets have been chosen. Table 1 shows the properties of these XML data sets.

To measure the performance variation based on the types of the XML data sets, we used several real XML data sets. SigmodRecord and Mondial as the real XML data sets were collected from the XML data repository at the University of Washington [22].

To measure the performance variation based on the types of the XML queries, we used different types of XPath queries. The list of XPath queries used in our experiments is shown in Table 2. The two letters in a query name indicate the data set on which the query is executed: "SI" denotes SigmondRecord and "MO" denotes Mondial. The number in a query name denotes the type of a query. The XPath queries with the query type 1–3 were used to test the simple path XML queries with different depths. The XPath queries with the query type 4 were used to test the XML queries having a predicate condition on the text content. The XPath queries with the query type 5 were used to test the XML queries having a predicate condition on the attribute value. The XPath queries with the query type 6 were used to test the twig pattern XML queries having a predicate condition at the end of lan XPath expression. The XPath queries with the query type 7 were used to test the twig pattern XML queries having a predicate condition in the middle of an XPath expression.

Table 1. The Properties of Different XML Data Sets

XML Data Set	SigmodRecord	Mondial
Size (KB)	467	1,743
Number of elements	11,526	22,423
Number of Attributes	3,737	47,423
Max Depth	6	5
Max fan Out	89	955

Table 2. XPath Query Sets of the Different XML Data Sets

XML Data Set	Query Name: XPath Expression
SigmodRecord	SI1:/SigmodRecord/issue/volume SI2:/SigmodRecord/issue/articles/article/title SI3:/SigmodRecord/issue/articles/article/authors/author SI4:/SigmodRecord/issue/articles/article/initPage[text()>"50"] SI5:/SigmodRecord/issue/articles/article/authors/author[@position="01"] SI6:/SigmodRecord/issue/articles/article[endPage/text()<"100"] SI7:/SigmodRecord/issue[number/text()>"2"]/articles/article/title
Mondial	MO1:/mondial/country/religions MO2:/mondial/country/city/population MO3:/mondial/country/province/city/name MO4:/mondial/country/languages[text()="Azeri"] MO5:/mondial/desert/located[@id="f0_37198"] MO6:/mondial/country/city[name/text()="Aarhus"] MO7:/mondial/country[name/text()="Germany"]/province/city/population

In the simulation environment, a set of XML nodes have been chosen randomly and uniformly from the XML data sets. The access level to these XML nodes is controlled by Accessibility Ratio. The Accessibility Ratio is the ratio of accessible XML nodes to entire XML nodes in the XML data set. In our experiments, we have changed this ratio between 10% up to 90% by steps of 20%.

Our proposed structure was compared with the SecNode structure proposed by [20, 21] since the SecNode structure is the only structure that broadcasts XML data securely in wireless mobile networks. To do this comparison, we need to implement the SecNode structure as well. In order to reach the most accuracy in experimental results and comparison, this simulation has been conducted in the most similar state of the running computer, operating system, programming language, implementation of algorithms and even type of variables. It should be noted that our proposed structure and access mechanism is able to process simple path XML queries having a wildcard and descendant axis, but the SecNode structure is not able to process these types of XML queries. Therefore, in this section, we did not show the experimental results for these types of XML queries. However, the experimental results for these types of XML queries are the same as other types of XML queries.

The performance metrics used in our experiments were the size ratio, the tuning time ratio, and the access time ratio. They are defined as Equations (1-3):

$$Size\ ratio = \frac{Size\ of\ the\ Encrypted\ XML\ Stream\ in\ the\ Secure\ G-node\ Structure}{Size\ of\ the\ Encrypted\ XML\ Stream\ in\ the\ SecNode\ Structure} \times 100 \quad (1)$$

$$Tuning\ time\ ratio = \frac{Number\ of\ buckets\ to\ read\ in\ the\ encrypted\ XML\ stream\ which\ uses\ the\ Secure\ G-node\ structure}{Number\ of\ buckets\ to\ read\ in\ the\ encrypted\ XML\ stream\ which\ does\ not\ use\ the\ SecNode\ structure} \times 100 \quad (2)$$

$$Access\ time\ ratio = \frac{Number\ of\ buckets\ to\ read\ in\ the\ encrypted\ XML\ stream\ which\ uses\ the\ Secure\ G-node\ structure}{Number\ of\ buckets\ to\ read\ in\ the\ encrypted\ XML\ stream\ which\ does\ not\ use\ the\ SecNode\ structure} \times 100 \quad (3)$$

It should be mentioned that the number of buckets read in both the active mode and doze mode are calculated to measure the access time ratio while the number of buckets read in the active mode is calculated to measure the access time ratio.

4.2 Experimental Results on XML Stream Size

Figure. 8 shows the size ratio on the different XML data sets with different accessibility ratios from 10% to 90%.

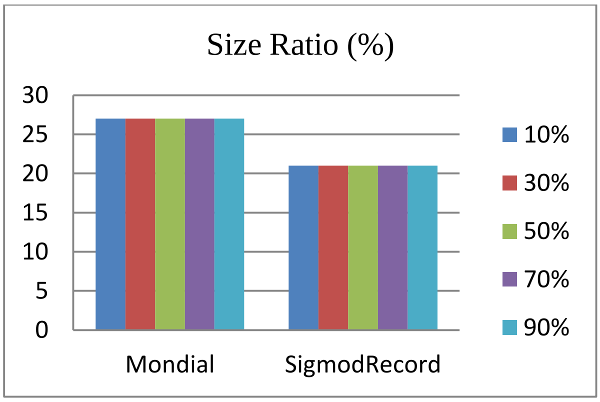

Figure 8. Size Ratio on the different XML Data Sets

As it is obvious in Figure. 8, the ratio size on the different XML data sets has changed. For the SigmodRecord data set, this ratio is about 21%, for the Mondial data set is about 27%. Therefore, the size of XML stream in our proposed structure is nearly half of the size of XML stream in the SecNode structure which this ratio is variable between 20% and 30%. It should be noted that the accessibility ratio does not affect the size of XML stream in both structures.

4.3 Experimental Results on Tuning Time

Figure. 9a-9b shows the ratio of tuning time in processing the different types of XML queries when the accessibility ratio is varied from 10% to 90% on the different XML data sets.

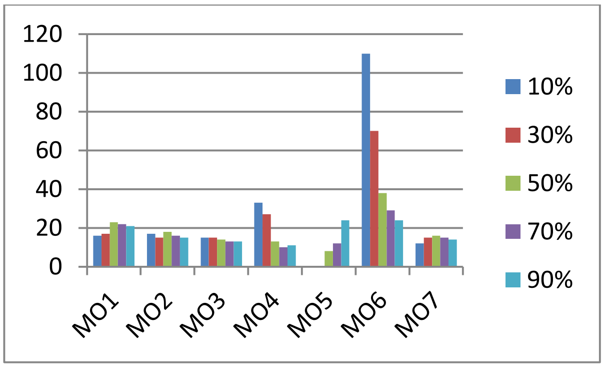

Figure 9. Ratio of Tuning Time on the Mondail Data Set

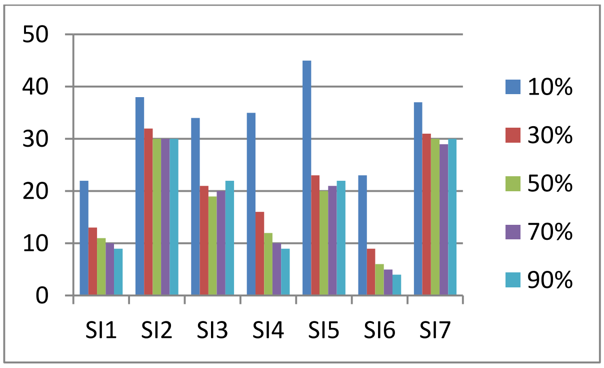

Figure.10. Ratio of Tuning Time on the SigmodRecord Data Set

From Figures. 9-10, it is clear that our proposed structure in comparison with the SecNode structure presents a better tuning time ever. However, this optimization is entirely dependent on the type of XML queries, the properties of XML data sets, and the accessibility ratio.

4.4 Experimental Results on Access Time

Figure. 11-12 shows the ratio of access time in processing the different types of XML queries when the accessibility ratio is varied from 10% to 90% on the different XML data sets.

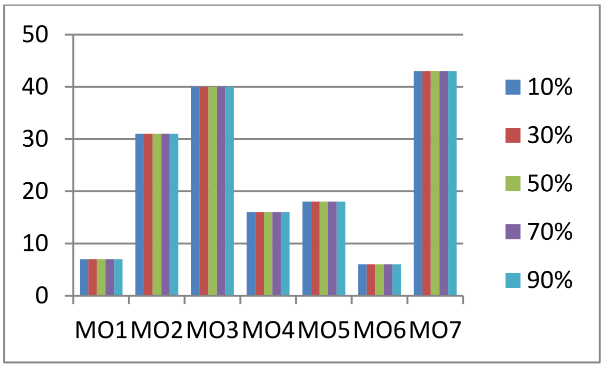

Figure.11. Ratio of Access Time on the Mondail Data Set

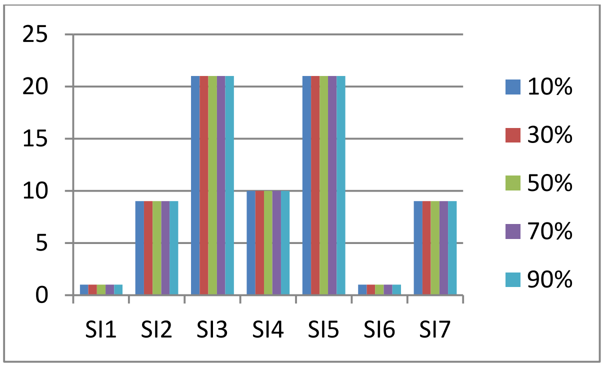

Figure.12. Ratio of Access Time on the SigmodRecord Data Set

From Figures. 10a-10b, it is clear that our proposed structure in comparison with the SecNode structure presents a better access time ever. However, this optimization is entirely dependent on the type of XML queries and the properties of XML data sets. Moreover, it is obvious that the different accessibility ratios are ineffective on the ratio of access time.

5. Conclusion

In this paper, a new structure was proposed to secure broadcast XML data in wireless mobile networks. One of the advantages of our proposed structure was its ability to process the different types of XML queries. In order to investigate the efficiency of our proposed structure, our proposed structure was compared with the SecNode structure which is of the newest studies in this research scope. The experimental results show that our proposed structure has averagely 60% advantage in tuning time and access time and also in XML stream size in comparison with the SecNode structure.

References

[1] T. Bray, J. Paoli, C.M. Sperberg-McQueen, E. Maler, and F. Yergeau, Extensible Markup Language (XML) 1.0 (5th Edition) W3C Recommendation," http://www.w3.org/TR/REC-xml/, 2008.

[2] J.P. Park, C.-S. Park, and Y.D. Chung, Energy and Latency Efficient Access of Wireless XML Stream, Journal of Database Management, Vol. 21, pp. 58-79, 2010.

[3] Y.D. Chung and M.H. Kim, An Index Replication Scheme for Wireless Data Broadcasting, Journal of Systems and Software, Vol. 51, pp. 191-199, 2000.

[4] C.-C. Lee and Y. Leu, Efficient Data Broadcast Schemes for Mobile Computing Environments with Data Missing, Information Sciences, Vol. 172, pp. 335-359, 2005.

[5] Y.D. Chung and J.Y. Lee, An Indexing Method for Wireless Broadcast XML Data, Information Sciences, Vol. 177, pp. 1931-1953, 2007.

[6] A.B. Boroujeni and M. Mirabi, A Novel Replication Strategy for Efficient XML Data Broadcast in Wireless Mobile Networks, Journal of Information Science and Engineering, Vol. 32, pp. 309-327, 2016.

[7] C.-S. Park, C.S. Kim, and Y.D. Chung, Efficient Stream Organization for Wireless Broadcasting of XML Data, 10th Asian Computing Science conference on Advances in Computer Science: Data Management on the Web, LNCS 3818, pp. 223-235, 2005.

[8] S.H. Park, J.H. Choi, and S. Lee, An Effective, Efficient XML Data Broadcasting Method in a Mobile Wireless Network, 17th International Conference on Database and Expert Systems Applications (DEXA 2006), LNCS 4080, pp. 358-367, 2006.

[9] J.P. Park, C.-S. Park, and Y.D. Chung, Lineage Encoding: An Efficient Wireless XML Streaming Supporting Twig Pattern Queries, IEEE Transactions on Knowledge and Data Engineering, Vol. 25, pp. 1559-1573, 2013.

[10] M. Mirabi, H. Ibrahim, and L. Fathi, PS+Pre/Post: A Novel Structure and Access Mechanism for Wireless XML Stream Supporting Twig Pattern Queries, Pervasive and Mobile Computing, Vol. 15, pp. 3-25, 2014.

[11] W. Sun, P. Yu, Y. Qing, Z. Zhang, and B. Zheng, Two-Tier Air Indexing for On-Demand XML Data Broadcast, 29th IEEE International Conference on Distributed Computing Systems, pp. 199-206, 2009.

[12] Y. Qin, H. Wang, and J. Xia, Effective Scheduling Algorithm for On-Demand XML Data Broadcasts in Wireless Environments, Australasian Database Conference (ADC '11), pp. 95-102, 2011.

[13] Y. Qin, W. Sun, Z. Zhang, P. Yu, and Z. He, Query-Grouping Based Scheduling Algorithm for On- Demand XML Data Broadcast, 4th International Conference on Wireless Communications, Networking and Mobile Computing, pp. 1-4, 2008.

[14] Y. Qin, W. Sun, Z. Zhang, and P. Yu, An Efficient Document-Split Algorithm for On-Demand XML Data Broadcast Scheduling, IET Conference on Wireless, Mobile and Sensor Networks (CCWMSN07), pp. 759-762, 2007.

[15] W. Sun, Y. Qin, P. Yu, and Z. Zhang, On-Demand XML Data Broadcast in Wireless Computing Environments, Third International Conference on Wireless Communications, Networking and Mobile Computing, pp. 3035-3038, 2007.

[16] Y. Qin, W. Sun, Z. Zhang, P. Yu, Z. He, and W. Chen, A Novel Air Index Scheme for Twig Queries in On- Demand XML Data Broadcast, 20th International Conference on Database and Expert Systems Applications (DEXA 2009), LNCS 5690, pp. 412-426, 2009.

[17] W. Sun, P. Liu, J. Wu, Y. Qin, and B. Zheng, An Automaton-Based Index Scheme for On-Demand XML Data Broadcast, 17th International Conference on Database Systems for Advanced Applications (DASFAA 2012), pp. 96-110, 2012.

[18] S.-H. Park, J.-H. Choi, and S. Lee, An Effective, Efficient XML Data Broadcasting Method in a Mobile Wireless Network,17th International Conference on Database and Expert Systems Applications (DEXA 2006), LNCS 4080, pp. 358-367, 2006.

[19] J.P. Park, C.-S. Park, and Y.D. Chung, Lineage Encoding: An Efficient Wireless XML Streaming Supporting Twig Pattern Queries, IEEE Transactions on Knowledge and Data Engineering, Vol. 25, pp. 1559-1573, 2013.

[20] L. Fathi, H. Ibrahim, and M. Mirabi, An Energy Conservation Indexing Method for Secure XML Data Broadcast in Mobile Wireless Networks, Pervasive and Mobile Computing, Vol. 13, pp. 125- 141, 2014.

[21] L. Fathi, H. Ibrahim, and M. Mirabi,An Air Indexing Method for Encrypted XML Data Broadcast in Mobile Wireless Network, 12th International ACM Workshop on Data Engineering for Wireless and Mobile Access, pp. 28-35, 2013.

[22] Xml Dataset, http://www.cs.washington.edu/research/xmldatasets, [Last Available 2016.8.10].

[23] E. Bertino and E. Ferrari, Secure and Selective Dissemination of XML Documents, ACM Transactions on Information and System Security, Vol. 5, pp. 290 -331, 2002.

[24] C. Geuer-Pollmann, XML Pool Encryption, 2002 ACM workshop on XML security, pp. 1-9, 2002.

[25] F.S. Gharehchopogh, R. Aliverdiloo, V. Banayi, A New Communication Platform for Data Transmission in Virtual Private Network, International Journal of Mobile Network Communications & Telematics (IJMNCT), Vol: 3, No: 2, pp. 1-9, 2013

[26] J. G. Lee and K.-Y. Whang, Secure Query Processing against Encrypted XML Data using Query-Aware Decryption, Information Sciences, Vol. 176, pp. 1928-1947, 2006.

[27] H. K. Ko, M.-J. Kim, and S. Lee, On the Efficiency of Secure XML Broadcasting, Information Sciences, Vol. 177, pp. 5505-5521, 2007.

[28] J. Gao, T. Wang, and D. Yang, XFlat: Query-Friendly Encrypted XML View Publishing, Information Sciences, Vol. 178, pp. 774-787, 2008.

[29] T.-K. Chang and G.-H. Hwang, Developing an Efficient Query System for Encrypted XML documents, Journal of Systems and Software, Vol. 84, pp. 1292-1305, 2011.

[30] S. Al-Khalifa, H.V. Jagadish, N. Koudas, J.M. Patel, D. Srivastava, and Y. Wu, Structural Joins: A Primitive for Efficient XML Query Pattern Matching, 18th International Conference on Data Engineering (ICDE'02, pp. 141-152), 2002.

[31] M. Mirabi, H. Ibrahim, N.I. Udzir, and A. Mamat, An Encoding Scheme based on Fractional Number for Querying and Updating XML Data, Journal of Systems and Software, Vol. 85, pp. 1831- 1851, 2012.