ISSN: 2350-0328
International Journal of Advanced Research in Science, Engineering and Technology
Vol. 3, Issue 7 , July 2016

Efficient stream structure for secure XMLdata broadcast in mobile wireless networks

Madeh Shokri, Mohammad Latifi, Hasan Aria, Elham Abbasi

Post Graduate Student, Department of Computer Engineering and Information Technology, University Nonprofit of Iranian, Tehran, Iran

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 propose a new structure for streaming XML data called CSSNn (Clustering and Sub-clustering Security Node) which guarantees confidentiality of the XML data in the wireless stream. We also define four indexes called NSP(n), NCP(n), Nearest Accessiable(NCNSn) and Nearest Accessiable(NCNSn) for the CSSNn structure based on the set of access authorizations specified in the original XML document in order to efficiently process the XML queries over the encrypted XML stream. The use of the CSSNn structure for secure XML data broadcast improves the performance of XML query processing in terms of tuning time and therefore reduces the power consumption at mobile clients.

KEYWORDS:Indexing, Mobile wireless broadcast, Data confidentiality, Access control, twig pattern matching.

I. INTRODUCTION

Wireless technology is rapidly gaining popularity. Many predict an emerging gigantic market where millions of mobile users will carry small, battery powered palmtops with wireless connections. Mobile users will be in constant need of stock information, traffic directions, local directory, weather information, etc. Wireless medium will be used as the “first mile” of the information highway that will disseminate massive amounts of information across the country. Organizing and accessing data on wireless communication channels1 are new challenges to the database and telecommunication communities.

Wireless information systems are currently realized in many areas as wireless technologies are rapidly gaining popularity [1-4]. In the systems, mobile clients carry small, battery powered hand-held devices such as PDA’s and mobile phones with limited data processing capabilities. In wireless systems, data broadcasting is widely used due to the bandwidth restriction of wireless communication [1,4]. The server disseminates data through a broadcast channel, and mobile clients listen on the channel and selectively retrieve information of their interests without sending requests to the server.

Wireless broadcasting is an effective information dissemination approach in the wireless mobile environment because of the following benefits: 1) the server can support a massive number of mobile clients without additional costs (i.e., scalability), 2) the broadcast channel is shared by many clients (i.e., the effective utilization of bandwidth), and 3) the mobile clients can receive data without sending request messages that consumes much energy [5,6].With the successful development and proliferation of various related technologies, XML (eXtensibleMarkup Language) [7] has been popularly used as a standard means to facilitate the representation and sharing of structured data across different information systems not only in the wired Internet environment but also in the wireless broadcast environment [8].

In public networks like a wireless broadcast network, the XML data must be encrypted before sending the XML data over the network in order to guarantee data confidentiality. In these networks, the delivery of the XML data to the mobile clients must obey a set of access authorizations specified on the XML data [9–12]. Mobile clients must only access the authorized parts of the XML data based on their access authorizations.

Several indexing methods are proposed to reduce the tuning time such as those proposed by [2, 4, 14, 15]. However, these indexing methods are not applicable to be used for the XML data since they are designed for flat data records

ISSN: 2350-0328
International Journal of Advanced Research in Science, Engineering and Technology
Vol. 3, Issue 7 , July 2016

identified by keys while the XML data are semi-structured and can be accessed by XML query languages like XPath [16] and XQuery [17].
Recently, several indexing methods have been proposed to selectively access the XML data over an XML stream in a mobile wireless broadcast network such as the methods proposed by [18–20]. However, these indexing methods 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 instead of accessing the whole of the XML stream. In order to selectively access the XML data over an encrypted XML stream, an efficient indexing method is required to specify the authorized parts of the XML data in the stream.
Let us consider an XML tree as shown in Figure 1. In Figure 1, 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 1, 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 ‘‘Border’’ is accessible to the mobile clients of group g2 but not to the mobile clients of groups g1 and g3.

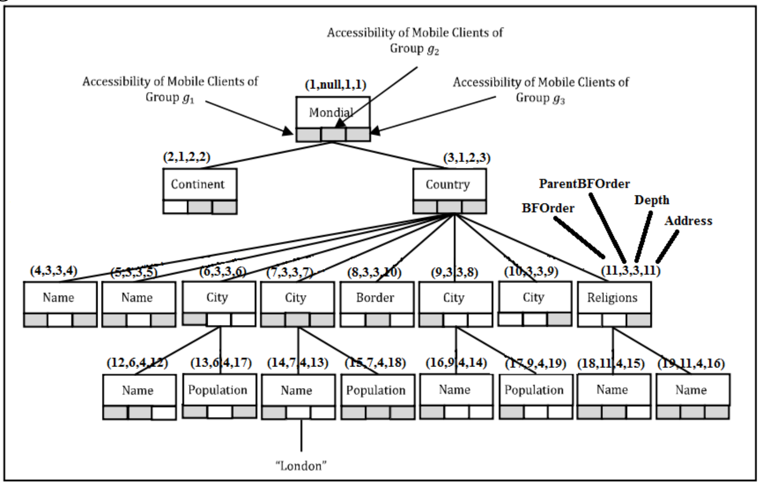

Fig. 1.Accessibility of different mobile clients over the XML tree

Figure 2 shows a broadcast stream of the XML tree illustrated in Figure 1. The broadcast server first encrypts all of the XML nodes separately with different keys and then organizes the encrypted XML nodes in the Address sequence and broadcasts them on the air. In Figure 2, it is assumed that all of the XML nodes are labelled with the BFOrderlabeling scheme since the XML nodes in the stream are encrypted separately and the structural relationships between the different XML nodes in the stream must be determined during the process of XML querying over the stream. The BFOrderlabeling scheme is able to determine all types of structural relationships between the different XML nodes. An example of an XML tree labelled by the BFOrderlabeling scheme is illustrated in Figure 1 where the BFOrder, ParentBFOrder, depth and Address values are assigned to each XML node in the XML tree.

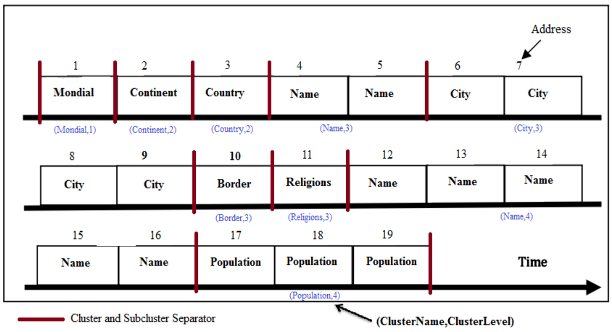

Fig. 2.An example of encrypted XML stream

ISSN: 2350-0328
International Journal of Advanced Research in Science, Engineering and Technology
Vol. 3, Issue 7 , July 2016

In the conventional approach to process an XML query over an encrypted XML stream, each mobile client first decrypts each encrypted XML node in the stream if it has the appropriate key for decryption and then searches the XML nodes in the stream satisfying the XML query path. Assume that a mobile client c1 which is a member of the group g1 listens to the broadcast channel and queries ‘‘/Mondial/Country/City/Name’’ before the root node is broadcast. The mobile client c1 can only decrypt the gray nodes in the stream shown in Figure 2 since it only has the set of keys for decrypting them based on the set of access authorizations specified for the members of the group g1 in the XML tree shown in Figure 1.

II. RELATED WORK

In this section, we review the existing research works on XML access control, XML data dissemination over the Internet, and XML data broadcast in mobile wireless networks.

In [23] a method for broadcasting indexing XML data securely on mobile wireless channel is provided. In this paper, a structure called SecNode that in fact an extension of DIX method proposed in [24] is. The structure of the fields KeyID (for node XML encryption for data confidentiality), Min (NCS), Min (NIS) (to jump to the node you want to access and query processing as well as reducing waiting time and also for controlling access to data XML) and Preorder, Postorder (to label nodes in XML query processing with pattern Twigg) uses. In [23] an authorization code (AC) defined based on different user groups to provide access to XML nodes. It can also be said [24] optimize the access time and tuning time to a certain extent.

Recently, several indexing methods have been proposed to selectively access the XML data over an XML stream in a mobile wireless broadcast network such as the methods proposed by [18–20]. The first method for streaming the XML data in a mobile wireless network is proposed by [18] which defines a new structure called an S-Node for an XML stream. In [18], indexes for the S-Node structure are defined based on the structural characteristics of an XML document such as the tag names and the paths of the XML nodes. The S-Node approach enables mobile clients to skip the irrelevant XML nodes during the XML query processing and therefore, improves the performance of the XML querying in terms of tuning time.

An XML streaming method is proposed by [20] using the concept of path summary [25]. By exploiting the concept of path summary for indexing the XML data in the stream, the redundant tag names in the XML stream are removed and therefore this method reduces the access time and tuning time in processing the XML queries.

A novel Distributed Indexing method for XML data called DIX is proposed by [19] which provides energy and latency efficient access to the XML data in the stream. The DIX structure contains a Location Path Information (LPI) and two indexes: Clone node Link (CL) and Foreign node Link (FL). The LPI is used by mobile clients to start the XML query processing over the stream at any time without the need to wait for the next broadcast cycle. The CL index is used to jump forward to the next candidate node which might be the query answer while the FL index is used to skip the irrelevant parts of the XML stream.

In general, some parts of an XML document are accessed multiple times by many mobile clients in a wireless broadcast channel. By considering this fact, Wu et al. [26] proposed two fragmentation methods called horizontal fragmentation method and threshold fragmentation method to fragment the hot parts of an XML document according to the access probability of each node in the XML document in order to efficiently broadcast the XML data in the push-based mode. In the horizontal fragmentation method, a level of the XML document is first selected as an optimal level and then the XML document is fragmented with the XML nodes located at the optimal level. Therefore, the core operation in the horizontal fragmentation method is to select an optimal level which is computed based on the access probability of the XML nodes. In the threshold fragmentation method, a weight value is assigned to each XML node of the XML document by exploiting the access probability of the XML nodes and then the XML node with the largest weight value is selected to generate a new fragmentation piece. This process is recursively performed on the fragmentation piece to generate new fragmentation pieces. Wu et al. [26] used the two-tier air indexing method proposed by [27] to index the fragmentation pieces. By exploiting this indexing method, the IDs of fragmentation pieces as well as their arrival times can be found efficiently on the air. Although the proposed fragmentation methods in [26] improve the performance of simple path XML query processing in terms of access time and tuning time but obtaining the access probabilities of the XML nodes before disseminating the XML data on the air is not an easy task.

In [28] a method based on the technique path summary [20] In the name of PS+Pre/Post suggested. It is actually a combination of technique and design label path summary Pre/Post are. Because this method uses the technique path summary can access time and time adjusted to remove the name tag improve extension. It also uses the Pre/Post labelling scheme can be a variety of Twig and conditional queries on XML data streams to be processed efficiently.

ISSN: 2350-0328
International Journal of Advanced Research in Science, Engineering and Technology
Vol. 3, Issue 7 , July 2016

III.PROPOSED XML STREAMING METHOD

To generate an XML stream in a secure manner, the contents of the XML data must be transformed from the original XML document into a suitable representation in such a way that they cannot be accessed by unauthorized parties in a wireless broadcast channel. In order to guarantee confidentiality of the XML data in the stream, they must be encrypted before broadcasting over the channel. The generated XML stream must also contain index information where indexes are constructed based on the set of access authorizations specified in the original XML document to support the XML query processing at mobile clients.

Figure 3 shows the structure of a CSSN for the node n ($CSSN_n$) in the stream which contains the following fields: KeyId, BFOrder, ParentBFOrder, Address, Path of Root-to-Currentnode, NSP, NCP, NearestAccessiable(NCNS), NearestAccessiable(NINS), Text, and AttList (Attributes List) of the node n.

KeyID (n)
BFOrder (n)
ParentBFOrder (n)
Address (n)
Path of Root-to-Currentnode (n)
NSP (n)
NCP (n)
NearestAccessiable($NCNS_n$)
NearestAccessiable($NINS_n$)
Text (n)
AttList (n) ={(name,value),...}

Fig. 3. Structure of $CSSN_n$ (Clustering and Sub-clustering Security Node

Below, we describe the proposed structure fields:

KeyID (n): The field KeyID (n) in the $CSSN_n$ is the key identifier used for encryption/decryption of the $CSSN_n$.

BFOrder (n): The fields BFOrder (n) in the $CSSN_n$ are the Breadth-First Order of the node n when the XML tree T is traversed in the Breadth-First Order.

ParentBFOrder (n): The fields ParentBFOrder (n) in the $CSSN_n$ are the Breadth-First Order parent node of the node n when the XML tree T is traversed in the Breadth-First Order.

Address (n): The fields Address (n) in the CSSNn are the order of nodes in the broadcast channel does specify.

Path of Root-to-Currentnode (n): The Path of Root-to-Currentnodeof the node n in the XML tree T is a sequence of node names (or tag names) from the root node r to the node n which are separated by ‘‘/’’.

For example, the Path of Root-to-Currentnode of the node ‘‘Name’’ with the BFOrder 12 (called node 12) in the XML tree illustrated in Figure 1 is the path ‘‘/Mondial/Country/City/Name’’.

The field Path of Root-to-Currentnode (n) in the $CSSN_n$ is used by the mobile clients to identify the exact path of the $CSSN_n$ in the stream in processing the XML queries.

NSP (Nearest Sub-cluster Pointer): The field NSP (n) in the $CSSN_n$ is nearest sub-cluster on the level (depth) next broadcast channel in the XML tree.

For example, the NSP of the node ‘‘Name’’ with the Address=4 in the XML tree illustrated in Figure 1 is Address=6.

Note that, if at the same level in the XML tree (broadcast channel), your target node is the last node in sub-cluster, Nsp amount will be equal to null.

NCP (Nearest Cluster Pointer): The field NCP (n) in the $CSSN_n$ is nearest cluster on the level (depth) next broadcast channel in the XML tree.

For example, the NCP of the node ‘‘City’’ with the Address=6 in the XML tree illustrated in Figure 1 is Address=12.

Definition 1 (Next Candidate Node). Node x is the next candidate node of the node z in the XML tree T if it satisfies the following three conditions:

- $BFOrder_x > BFOrder_z$.
- $Address_x > Address_z$
- Node x has the same Path of Root-to-Currentnode as the node z.

For example, the node with Address=13 (i.e. node ‘‘Name’’) with the Path of Root-to-Currentnode ‘‘/Mondial/Country/City/Name’’ is the next candidate node of node with Address=12 (i.e. node ‘‘Name’’) with the Path of Root-to-Currentnode ‘‘/Mondial/Country/ City/Name’’ in the XML tree illustrated in Figure 1.

ISSN: 2350-0328

International Journal of Advanced Research in Science, Engineering and Technology

Vol. 3, Issue 7 , July 2016

Definition 2 (Accessibility Code). 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 $AC_e = a_1a_2 \dots a_m$ where

$$\forall 1 \le i \le m, \quad a_i = \begin{cases} 1, & \text{if the } i\text{th user group in } UG, g_i, \text{ has access to the node } e \\ 0, & \text{otherwise} \end{cases}$$

and m is the total number of user groups in UG.
For example, the node with Address=10 (i.e. node ‘‘Border’’) in the XML tree illustrated in Figure 1 can only be accessed by mobile clients who are the members of the group g2. Therefore, the accessibility code of the node 10 is 010 ($AC_{node\ 10}$ = 010).
Definition 3 (Next Candidate Node Set). Let node x be the next candidate node of node z, AC_x be the accessibility code of node x, and Add_x be the address (arrival time) of node f in the stream. The next candidate node set of node z ($NCNS_z$) is an ordered set of the next candidate nodes of node z defined as follows:

$$NCNS_z=\{(AC_x,Add_x) \mid AC_x \neq 0^m \text{ (m occurrences of 0)}\}$$

where m is the total number of user groups in UG and the pairs of (AC_x , Add_x) are ordered in ascending based on their address (Add).
For example, the next candidate node set of node with Address=17 (i.e. node ‘‘Population’’) in the XML tree illustrated in Figure 1 is as follows: $NCNS_{node\ 17}$ = {(111, 18), (100, 19)}.
NearestAccessiable($NCNS_n$): The field NearestAccessiable($NCNS_n$) in the $CSSN_n$ is next available set of candidate nodes closest to the current node specifies.
For example, the NearestAccessiable(NCNS) of node with Address=4 (i.e. node ‘‘Name’’) in the XML tree illustrated in Figure 1 is as follows: NearestAccessiable($NCNS_{node\ 4}$) = {(101, 5)}.
Definition 4 (Next Irrelevant Node). Node x is a next irrelevant node of the node z in the XML tree T if it satisfies the following three conditions:

- $BFOrder_x > BFOrder_z$
- $Address_x > Address_z$
- Node x has the same depth as node z;
- The Path of Root-to-Currentnode of the parent node of node x is different from that of node z.

For example, node with Address=15 (i.e. node ‘‘Name’’) is the next irrelevant node of node with Address=12 (i.e. node ‘‘Name’’) in the XML tree illustrated in Figure 1, since the Path of Root-to-Currentnode of the parent node of node 12 (i.e. ‘‘/Mondial/Country/City’’) is different from that of node 15 (i.e. ‘‘/Mondial/Country/Religions’’).
Definition 5 (Next Irrelevant Node Set). Let node x be the next irrelevantnode of node z, AC_x be the accessibility code of node x, and Add_x be the address (arrival time) of node f in the stream. The next irrelevant node set of node z ($NINS_z$) is an ordered set of the next irrelevant nodes of node z defined as follows:
$NINS_z=\{(AC_x,Add_x) \mid AC_x \neq 0^m \text{ (m occurrences of 0)}\}$
where m is the total number of user groups in UG and the pairs of (AC_x , Add_x) are ordered in ascending based on their address (Add).
For example, the next irrelevantnode set of node with Address=12 (i.e. node ‘‘Name”) in the XML tree illustrated in Figure 1 is as follows: $NINS_{node\ 12}$ = {(110,15),(111,16)}.
NearestAccessiable($NINS_n$): The field NearestAccessiable($NINS_n$) in the $CSSN_n$ is next available set of irrelevant nodes closest to the current node specifies.
For example, the NearestAccessiable(NINS) of node with Address=12 (i.e. node ‘‘Name’’) in the XML tree illustrated in Figure 1 is as follows: NearestAccessiable($NINS_{node\ 12}$) = {(110,15),(111,16)}.
Text (n): The fields Text (n) in the $CSSN_n$ represent the content of the node n in the XML tree T.
AttList(n): The fields AttList (n) in the $CSSN_n$ represent the list of attributes of the node n in the XML tree T. The field AttList (n) contains a set of pairs (name, value) representing the name and value of each attribute of the node n.

IV. XML QUERY PROCESSING OVER THE ENCRYPTED XML STREAM

In this section, we explain how a mobile client can process the different types of XML queries over an encrypted XML stream using the CSSN structure.

A. Processing of simple path XML queries

To process the simple path XML queries over the encrypted XML stream, we act as follows.

ISSN: 2350-0328
International Journal of Advanced Research in Science, Engineering and Technology
Vol. 3, Issue 7 , July 2016

An example of a simple path query processing over the encrypted XML stream using the field NearestAccessiable(NCNS) and the field NCP and the field Address is illustrated in Example 1.
Example 1. As shown in Figure 4, assume that the mobile client c1 which is a member of the group g1 listens to the broadcast channel when the node 5 (i.e. node ‘‘Name’’) is on the air and submits the XML query ‘‘/Mondial/Country/City/Name’’.

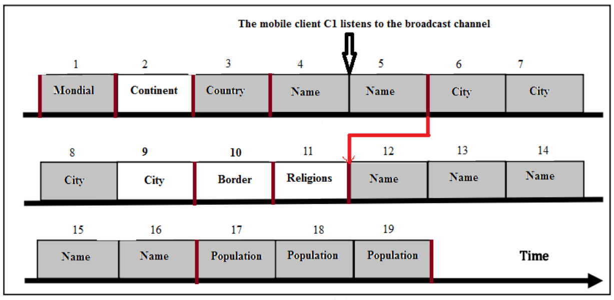

Fig. 4. An example of simple path query processing using the fieldNearestAccessiable(NCNS) and the field NCP and the field Address

The mobile client c1 can decrypt node 5 since the KeyID of node 5 matches one of the Key IDs of the mobile client c1. After decrypting node 5, the mobile client c1 computes the depth of node 5 using the field Path of Root-to-Currentnode of node 5. Since the depth of node 5 is different from that of the submitted XML query, the mobile client c1 continues to read the encrypted CSSNs in the stream. Then using the Field NCP, mobile client went into doze mode until the first node level 4 (i.e. Address=12 or node “Name”) over the broadcast channel, the mobile client to active mode and the time it gets downloaded. After reading and decrypting node 12, the mobile client c1 compares the field Path of Root-to-Currentnode of node 12 to the submitted XML query path. The mobile client c1 finds that node 12 satisfies the XML query and therefore, it adds node 12 to the set response. By exploiting the field NearestAccessiable(NCNS) of node 12 (i.e. NearestAccessiable(NCNS node 12 = {(100, 13)})), the mobile client c1 can predict the arrival time of the next accessible candidate node (i.e. node 13) on the air. The mobile client c1, node 13 satisfies the XML query and therefore, it adds node 13 to the set response. By exploiting the field NearestAccessiable(NCNS) of node 13 (i.e. NearestAccessiable(NCNS node 13 = {(100, 14)})), the mobile client c1 can predict the arrival time of the next accessible candidate node (i.e. node 14) on the air. The mobile client c1, node 14 satisfies the XML query and therefore, it adds node 14 to the set response.
An example of processing the simple path XML queries over the encrypted XML stream using the field fieldNearestAccessiable(NINS) and the field NCP and the field Address is illustrated in Example 2.
Example 2. As shown in Figure 5, assume that the mobile client c_1 which is a member of the group g1 listens to the broadcast channel when the node 5 (i.e. node ‘‘Name’’) is on the air and submits the XML query ‘‘/Mondial/Country/Religions/Name’’.

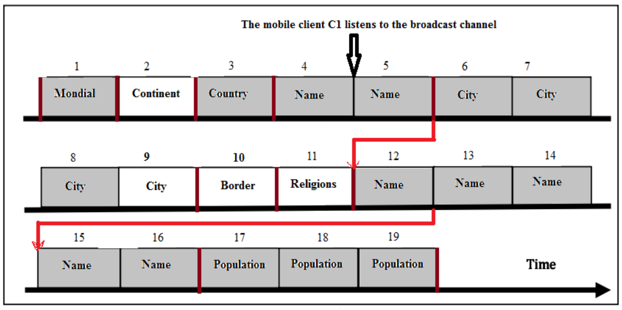

Fig. 5. An example of simple path query processing using the field NearestAccessiable(NINS) and the field NCP and the field Address

The mobile client c1 can decrypt node 5 since the KeyID of node 5 matches one of the Key IDs of the mobile client c1. After decrypting node 5, the mobile client c1 computes the depth of node 5 using the field Path of Root-to-Currentnodeof node 5. Since the depth of node 5 is different from that of the submitted XML query, the mobile client

ISSN: 2350-0328
International Journal of Advanced Research in Science, Engineering and Technology
Vol. 3, Issue 7 , July 2016

c1 continues to read the encrypted CSSNs in the stream. Then using the Field NCP, mobile client c_1 went into doze mode until the first node level 4 (i.e. Address=12 or node "Name") over the broadcast channel, the mobile client to active mode and the time it gets downloaded. After reading and decrypting node 12, the mobile client c1 compares the field Path of Root-to-Current node of node 12 to the submitted XML query path. After decrypting node 12, the mobile client c_1 computes the depth of the node 12 using the field Path of Root-to-Currentnode of node 12. The mobile client c_1 finds that the depth of node 12 is equal to the depth of the given XML query but the Path of Root-to-Currentnode of node 12 is different from the path of the XML query.

Since none of the accessible nodes having the same Path of Root-to-Currentnode as node 12 can be the result of the XML query, the mobile client c_1 can efficiently skip the irrelevant accessible nodes in the stream by using the field NearestAccessiable(NINS) of node 12 (i.e. NearestAccessiable(NINS $_{\text{node 12}}$ = {(110, 15), (111, 16)})). By exploiting the field NearestAccessiable(NINS) of node 12, the mobile client c_1 can predict the arrival time of node 15 on the air. Therefore, the mobile client c_1 switches to the doze mode to conserve its battery power. When node 15 is on the air, the mobile client c_1 wakes up and reads node 15. The mobile client c_1 finds the first answer at node 15 since it has the same Path of Root-to-Currentnode as the XML query.

B. Processing of twig pattern XML queries

A twig pattern XML query is an XML query with two or more XPath expressions which contain predicate conditions. In general, a predicate condition can appear at the end of an XPath expression (e.g. ''/Mondial/Country/City[Name/text()= ''Sydney'']'') or in the middle of an XPath expression (e.g. ''/Mondial/Country/ City[Name/text()=''London'']/Population'').

Typically, a twig pattern XML query having a predicate condition at the end of an XPath expression can be decomposed into two XML queries. For example, the twig pattern XML query ''/Mondial/Country/ City[Name/text()=''London''])'' can be decomposed into two XML queries''/Mondial/Country/City''and''/Mondial/Country/City/Name[text()=''London'']''. The second XML query is a simple path XML query having a predicate condition over the text content. We use the method Simple Path Query Processing in order to process the two decomposed XML queries. After finding the results of both the XML queries, we use the BFOrder, ParentBFOrder, and Depth values of the nodes in the results of the two XML queries to determine the results of the twig pattern XML query. This process is done by exploiting the following properties:

Property 1. An XML node α labeled by ($BFOrder_\alpha$,$ParentBFOrder_\alpha$, $Depth_\alpha$,$Address_\alpha$) is the ancestor node of an XML node β labeled by ($BFOrder_\beta$, $ParentBFOrder_\beta$, $Depth_\beta$,$Address_\beta$) if and only if $BFOrder_\alpha<BFOrder_\beta$ and $Depth_\alpha< Depth_\beta$ and $Address_\alpha< Address_\beta$.

For example, in Figure 1, the node ''Country'' with the label (3, 1, 2, 3) is the ancestor node of the node ''Population'' with the label (15, 7, 4, 18) since 3 < 15 and 2 < 4 and 3 < 18.

Property 2. An XML node α labeled by ($BFOrder_\alpha$,$ParentBFOrder_\alpha$, $Depth_\alpha$,$Address_\alpha$) is the parent node of an XML node β labeled by ($BFOrder_\beta$, $ParentBFOrder_\beta$, $Depth_\beta$,$Address_\beta$) if and only if $BFOrder_\alpha<BFOrder_\beta$ and $BFOrder_\alpha$ = $ParentBFOrder_\beta$ and $Depth_\alpha$ = $Depth_\beta$ -1 and $Address_\alpha< Address_\beta$.

For example, in Figure 1, the node ''City'' with the label (9, 3, 3, 8) is the parent node of the node ''Name'' with the label (16, 9, 4, 14) since 9 < 16 and 9=9 and 3 = 4-1 and 8 < 14.

V. CONCLUSION

In this paper, we have proposed a wireless streaming method for XML data whichsupports energy-efficient processing of queries over a stream in mobile clients. Particularly, our method is effective for query processing on time critical data. That is, our method provides the fast response time to the mobile client.We defined a unit structure of an XML stream called CSSN (Clustering and Sub-clustering Security Node) which guarantees confidentiality of the XML nodes in the stream. The CSSN structure contains the field Path of Root-to-Currentnode to process the simple path XML queries. Four indexes NSP, NCP, NearestAccessiable(NCNS), NearestAccessiable(NINS) were defined for the CSSN structure to reduce the tuning time in processing te XML queries. We also proposed a method to process twig pattern XML queries having predicate conditions by decomposing the twig pattern XML query into several simple path XML queries.

REFERENCES

[1] Acharya, S., et al.: Broadcast Disks: Data Management for Asymmetric Communication Environments. Proc. of ACM SIGMOD Conf. (1995) 199–210.

[2] Chung, Y.D., Kim,M.H.: An Index Replication Scheme for Wireless Data Broadcasting. Journal of Systems and Software, 51(3) (2000) 191–199.

[3] Imielinski, T., Badrinath, B.R.: Data Management for Mobile Computing. SIGMOD RECORD, 22(1) (1993).

[4] Imielinski, T., et al.: Data on Air: Organization and Access. IEEE Trans. on Knowledge and Data Eng., 9(3) (1997).

[5] T. Imielinski, S. Viswanathan and B. Badrinath, "Energy efficient indexing on air," Proc. ACM SIGMOD Conf., pp. 25-36, 1994.

[6] Y. D. Chung, S. Yoo, and M. H. Kim, "Energy- and Latency-Efficient Processing of Full-Text Searches on a Wireless Broadcast Stream," IEEE Trans. Knowledge and Data Eng., vol. 22, no. 2, 2010.

[7] W3C Recommendation. (2006). Extensible Markup Language (XML) 1.0 (4th ed.). Retrieved from http://www.w3.org/XML.

[8] Fong, J., & Wong, H. K. (2004). XTOPO: An XML-Based Topology for Information Highway on the Internet. Journal of Database Management, 15(3), 18–44.

[9] J.-G. Lee, K.-Y. Whang, Secure query processing against encrypted XML data using query-aware decryption, Information Sciences 176 (13) (2006) 1928–1947.

[10] H.-K. Ko, M.-J. Kim, S. Lee, On the efficiency of secure XML broadcasting, Information Sciences 177 (24) (2007) 5505–5521.

[11] J. Gao, T. Wang, D. Yang, XFlat: query-friendly encrypted XML view publishing, Information Sciences 178 (3) (2008) 774–787.

[12] Y.D. Chung, S. Yoo, M.H. Kim, Energy and latency efficient processing of full-text searches on a wireless broadcast stream, IEEE Transactions on Knowledge and Data Engineering 22 (2) (2010) 207–218.

[13] M.-S. Chen, K.-L. Wu, P.S. Yu, Optimizing index allocation for sequential data broadcasting in wireless mobile computing, IEEE Transactions on Knowledge and Data Engineering 15 (1) (2003) 161–173.

[14] A. Berglund, S. Boag, D. Chamberlin, M.F. Fernández, M. Kay, J. Robie, J. Siméon, XML Path Language (XPath) 2.0, second ed., 2010. http://www.w3.org/ TR/xpath20/.

[15] S. Boag, D. Chamberlin, M.F. Fernández, D. Florescu, J. Robie, J. Siméon, XQuery 1.0: An XML Query Language, second ed., 2010. http://www.w3.org/ TR/xquery/.

[16] S. Al-Khalifa, H.V. Jagadish, N. Koudas, J.M. Patel, D. Srivastava, Y. Wu, Structural joins: a primitive for efficient XML query pattern matching, in: Proceedings of the 18th International Conference on Data Engineering, ICDE'02, San Jose, California, USA, 2002, pp. 141–152.

[17] C. Zhang, J. Naughton, D. DeWitt, Q. Luo, G. Lohman, On supporting containment queries in relational database management systems, ACM SIGMOD Record 30 (2) (2001) 425–436.

[18] C.-S. Park, C.S. Kim, Y.D. Chung, Efficient stream organization for wireless broadcasting of XML data, in: Proceedings of the 10th Asian Computing Science Conference on Advances in Computer Science: Data Management on the Web, in: LNCS, vol. 3818, Kunming, China, 2005, pp. 223–235.

[19] J.P. Park, C.-S. Park, Y.D. Chung, Energy and latency efficient access of wireless XML stream, Journal of Database Management 21 (1) (2010) 58–79.

[20] S.-H. Park, J.-H. Choi, S. Lee, An effective, efficient XML data broadcasting method in a mobile wireless network, in: Proceedings of the 17th International Conference on Database and Expert Systems Applications, DEXA 2006, in: LNCS, vol. 4080, Krakow, Poland, 2006, pp. 358–367.

[21] J.-G. Lee, K.-Y. Whang, W.-S. Han, I.-Y. Song, The dynamic predicate: integrating access control with query processing in XML databases, The VLDB Journal 16 (3) (2007) 371–387.

[22] S. Cho, S. Amer-Yahia, L.V.S. Lakshmanan, D. Srivastava, Optimizing the secure evaluation of twig queries, in: Proceedings of the 28th International Conference on Very Large Data Bases, Hong Kong, China, 2002, pp. 490–501.

[23] L. Fathi, H. Ibrahim, M. Mirabi," An energy conservation indexing method for secure XML data broadcast in mobile wireless networks", Pervasive and Mobile Computing 13 (2014) 125–141.

[24] J. P. Park, C.-S. Park, Y. D. Chung, Energy and latency efficient access of wireless XML stream, Journal of Database Management 21 (1) (2010) 58–79.

[25] R. Goldman, J. Widom, DataGuides: enabling query formulation and optimization in semistructured databases, in: Proceedings of the 3rd International Conference of Very Large Data Bases, Tokyo, Japan, 1997, pp. 436–445.

[26] J. Wu, P. Liu, L. Gan, Y. Qin, W. Sun, Energy-conserving fragment methods for skewed XML data access in push-based broadcast, in: Proceedings of the 12th International Conference on Web-Age Information Management, WAIM 2011, Wuhan, China, 2011, pp. 590–601.

[27] W. Sun, P. Yu, Y. Qing, Z. Zhang, B. Zheng, Two-tier air indexing for on-demand XML data broadcast, in: Proceedings of the 29th IEEE International Conference on Distributed Computing Systems, Montreal, Quebec, Canada, 2009, pp. 199–206.

[28] M. Mirabi, H. Ibrahim, L. Fathi, "PS+Pre/Post: A novel structure and access mechanism for wireless XML stream supporting twig pattern queries", Pervasive and Mobile Computing 15 (2014) 3–25