International Journal of Database Theory and Application
Vol.8, No.4 (2015), pp.215-230
http://dx.doi.org/10.14257/ijdta.2015.8.4.22

Implementation and Verification of Dissemination of Tree Structure Data using Signature Scheme for XML Data

Vivek N. Waghmare and Ravindra C. Thool

Abstract

With wide spread use of digital data communication on networks, security of data has become an important issue which provides integrity and confidentiality of content that ensures the distribution *of information appropriately. While dealing with encoded content in the context of XML, its hierarchical tree structure has different level of confidentiality and integrity for various portion of the same content. Thus it imposes the need of dissemination approach specifically tailored for XML data to address the issues of efficiency and scalability. However these characteristics must be achieved without compromising the security and privacy of contents. The goal of efficient proposed Signature scheme for secure & selective distribution of XML content is to provide uniform platform for different data representation for secure and efficient availability of data. Proposed Signature Scheme is used to detect the change of content at information providers to discover and deliver new content to users. This can be achieved using structural properties of hierarchical tree structure data model with multicast topology approach.*

***Keywords**: Confidentiality, Hierarchical Tree Structure Data, Integrity, XML*

1. Introduction

The main problem of content dissemination in an enterprise environment is to investigate various dissemination techniques [1, 2]. The transformation and growth of enterprise networks into more dynamic frame works rather than just a passive repository of contents has increase ubiquity of services and has contributed significantly to solve more complex problem. The evolution of XML and its impact on the data models has made the Document Object Model (DOM) [3] available for content representation. Enterprise computing paradigms have been using the XML DOM as the primary benchmark for data representation. Web services in an intra-enterprise network and across enterprise networks being adopted as the components for the distributed computing. These web services are primarily XML based services. Recent developments in content network appliances based technology can be used at the network level for efficiently filtering the contents before distributing them to interested parties over distributed systems.

Efficiency and scalability needs to be provided by ensuring the security of contents and privacy of the parties acquiring and disseminating the contents [14]. It is useless to provide high-bandwidth content distribution systems if integrity of the disseminated contents is not assured or the property of the contents not protected. Such problems are further complicated while dealing with contents encoded in XML [14], because of the hierarchical organization of the content as well as different confidentiality and integrity requirements which may exist for different portions of the same contents. Thus need a dissemination approach specifically tailored to XML that addresses the issues of security, privacy and scalability in a holistic manner.

Relevant requirements for such a dissemination approach include the following.

a. Secure Dissemination of Content
b. Selective Dissemination of Content

ISSN: 2005-4270 IJDTA

a. Secure Dissemination of Content [5, 14]:

Dissemination of hierarchical data model requires different security for different portion of data.

Requirement for such a dissemination approach include the following [5]:

i. **Access Control:** To prevent unauthorized users to infer sensitive information through the data they authorized to access [15].
ii. **Data Integrity:** Not only the integrity of the data must be verifiable by the user, but also any compromise to the data must be exactly determined.
iii. **Data Confidentiality:** A user receives only that information that user is allowed to access, according to access control policies and not able to infer any information that user is not authorized to access [16].

b. Selective Dissemination of Content [4]:

Selective Dissemination of Information (SDI) was the first concept described by H. P. Luhn [4] of IBM in the 1950's. Many companies and government organizations were providing this service in the 50's and 60's, which allowed distribution of items recently published in abstract journals to be routed to individuals who are likely to be interested in the contents. SDI is a concept that originates in the roots of computer science.

SDI systems match new information items to express user interest [4]. Fundamental to such system is the task of detecting the change of content at information providers because this is a precondition to discover and deliver new contents to users. Data not accessible to user but belongs to complete data set is an extraneous data. Flow of this extraneous data to a user may leak information, even when data is encrypted [14]. Therefore, data sharing among multiple parties require secure and selective dissemination of data without any leakage of information. Removing this extraneous data is complicated when data is organized in hierarchical model. SDI may provide a solution to this problem [4].

2. Basic Concept

XML (eXtensible Markup Language) [6] is standard for document interchanges languages on Web. It is a platform for application unification and management on the Internet. XML document contain information of different sensitivity degrees that should be shared by possible large users. Each tagged elements have zero or more elements and attributes, and contain textual information. Elements can be nested at any depth in the document structure [6]. The relation between parent and child nodes is represented as directed edges, with edges directed from parents to child.

There are two types of tags used in XML, the start tag, at the beginning of the element, with the form <*tag-name*>, and the end tag, at the end tag, at the end of the element, with the form </*tag-name*>.

Let *D* be a document and *T* be a DOM [3] tree representation of *D.*

T (V, E),

Where, *V* is the set of vertices and *E* is the set of edges.

T is a nonlinear, acyclic data structure.

Content be content only at *x* and Content$_x$ be the content specific to *x* and not of other nodes.

The Dissemination of a Document exploits following Structural properties of XML data [7]:

a. Order preserving XML data i.e. nodes *x* and *y* have an order among them in *D*.
b. Unit of data access is sub-tree representation of a subdocument. The smallest unit is a node.
c. Any element and its corresponding subdocument are accessible by themselves or by a sub-tree rooted at any of their ancestor.

3. Related Work

Approach suggested by Bertino and Ferrari supports access control in both pull and push based distribution of data [8, 9]. Information pull is based on authorization. Consumer sends request to source for XML document. When consumer submits an access request then access control system checks authorization of consumer. Based on this authorization, consumer is returned a view of the requested document that contains all and only those portions. When no authorizations are found then, access is denied. Information push approach is used for distributing documents to users which based on broadcast data to clients [14]. In this case, different users may have privileges to see different, selected portions of the same document. Thus, different views of same document are sent to different consumer. Example, in case of a newsletter sent once a week to all users. Different users have different privilege to access different, selected portion of same document, supporting an information push approach for generating different physical views of the same document and sending them to proper users. The main problem with Information pull and Information push approach is number of views becomes large and such approach cannot be practically applied. Gladney and Lopspiech proposed solution for above problem which is mainly based on, Multilevel Encryption [5, 8 and 10]. In multilevel encryption, different portions of same document are encrypted with different keys and same encrypted copy is broadcast to all subjects.

Issues related to multilevel encryption are as follows [8, 10]:

a. Which and how many keys should be distributed to which subjects?
b. How to securely and efficiently distribute keys?
c. How to encrypt document?

Solutions for these issues are,

1. Encryption of document according to specified access control policies.
2. According to policies apply key, therefore number of policies equal to number of keys.

Merkle proposed a digital signature scheme [5, 8, 10, and 14] based on a secure conventional encryption function over a hierarchy (tree) of document fragments. Buldas and Laur [14] have also found that Merkle trees are binding (integrity-preserving) but not hiding (confidentiality-preserving) the information. The use of commutative hash operations to compute the Merkle hash signature which prevents leakage related to the ordering among the siblings. However it cannot prevent the leakage of signatures at the node [10, 14] and to resolve structural relationships with its descendants or ancestors. Moreover, one-way accumulation is very expensive as compared to one-way hash operation. The Merkle hash technique has been widely used in data authentication [10]. Devanbu [17], used the Merkle hash technique for authenticating XML data. Bertino [5, 8, 10] proposed a technique based on the Merkle hash technique for selective dissemination of XML data in a third party distribution framework [4]. This technique is not scalable and does not remove extraneous data. It is sensitive to data tampering attack and inference attack. Kocher proposed to use Merkle hash trees for distribution to third parties. For secure multicast, Perrig uses static data ordering over symmetric encryption Chatvichienchai and Iwaihara[8, 10] proposed mechanisms for secure updates, without leading to information leakages. However such mechanism does not address the problem of information leakages during verification of integrity of partial XML documents [5].

4. Design Issues

Proposed signature scheme is based on notion of Encrypted Post Order Numbers (EPON) [14], which facilitates efficient dissemination of selected portion of the content. By using structure based routing (SBR) scheme, it prevents information leakage and assures that delivered content are according to access control policies of the user. The structure based routing framework facilitates the dissemination of contents with varying degrees of confidentiality and integrity in a network. Figure 1 represents block diagram of implemented Proposed Signature Scheme. The proposed Signature Scheme approach for enhancing the security for XML document is mainly divided into three Modules..,

a. Generation of EPON Value [14].
b. Document Encoding and Encryption.
c. Structure Based Routing.

a. Generation of EPON Value [5, 13, 14]

In this, take input as a XML file and create DOM tree for XML file. Then assign post order number to each node in the tree. Generate sorted random number and combine with post order number. These combined numbers are given as input to order preservation technique which creates encrypted post order number for tree as shown in Figure 1 (a). EPON values can be Generate by using three techniques.

1. Post Order Numbering

Let p_x be post order number [1, 2] assigned to each node in tree according to Post order traversal of tree. The highest PON is $|V|$ and lowest is 1. If z is the parent of left child x and right child y, then $p_y = p_x+1$ *and* $p_z = p_y+1$. PON of left most child of T is 1.

Properties of PONs are:

- p_x uniquely refers to x and subdocument D_x in.
  This property is used to identify and extract a specific sub-document in a document.

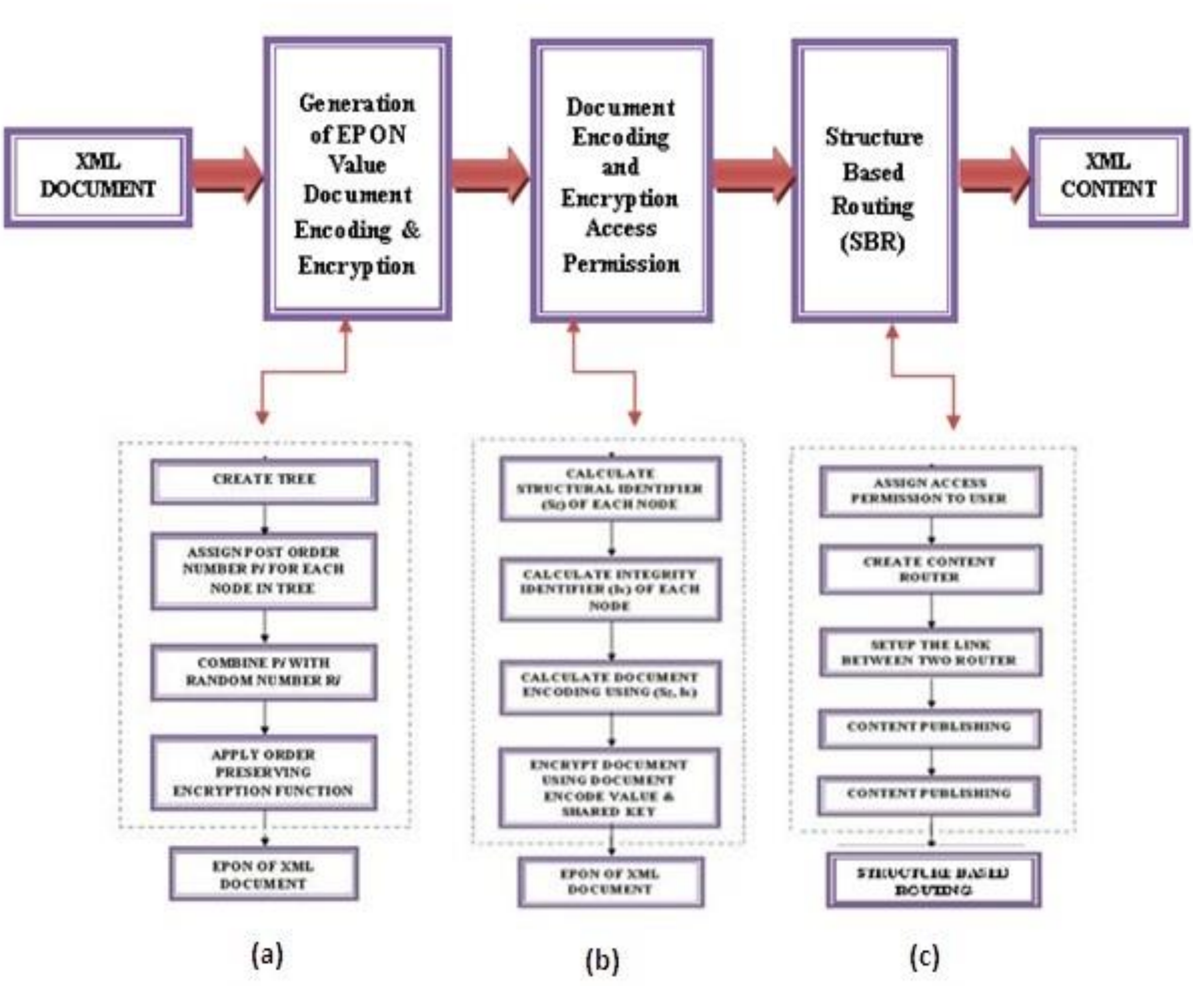

XML DOCUMENT
Generation of EPON Value Document Encoding & Encryption
Document Encoding and Encryption Access Permission
Structure Based Routing (SBR)
XML CONTENT
CREATE TREE
ASSIGN POST ORDER NUMBER Pi FOR EACH NODE IN TREE
COMBINE Pi WITH RANDOM NUMBER Ri
APPLY ORDER PRESERVING ENCRYPTION FUNCTION
EPON OF XML DOCUMENT
CALCULATE STRUCTURAL IDENTIFIER (Sc) OF EACH NODE
CALCULATE INTEGRITY IDENTIFIER (Ic) OF EACH NODE
CALCULATE DOCUMENT ENCODING USING (Sc, Ic)
ENCRYPT DOCUMENT USING DOCUMENT ENCODE VALUE & SHARED KEY
EPON OF XML DOCUMENT
ASSIGN ACCESS PERMISSION TO USER
CREATE CONTENT ROUTER
SETUP THE LINK BETWEEN TWO ROUTER
CONTENT PUBLISHING
CONTENT PUBLISHING
STRUCTURE BASED ROUTING
(a)
(b)
(c)

Figure 1. Block Diagram (a) Encrypted Post Order Number Generation (b) Encoding and Encryption of Document, (c) Structure Based Routing

- Let *z* be parent of x,
  then $p_x < p_z$.
  It is the basis for reasoning about the relation between the parent and a child.
- Let p^x_{lowest} the lowest PON of any element in the subdocument D_x.
  let *u:* descendent of x,
  then $p^x_{lowest} \leq p_u \leq p_x$..

It imposes a lower and upper bound on the possible PON of any element in a sub tree *x*. It is used to determine if there any swapping among siblings in the received subdocument.

2. Generation of Encrypted Post Order Numbering

Let $\{p_1, p_2... p_n\}$ be a set of PONs for an XML document. Each p_i with i = 1, 2… n is combined with a unique random number r_i. The combined values are then encrypted by using an order preserving encryption function. The resulting set of numbers is the set of EPONs, is denoted by e_x. The random value associated with a PON follows strictly increasing order, with the lowest random value being associated with the lowest PON. In this, encryption process encrypts these numbers in such a way that they preserve order among entities.

3. Order Preserving Encryption Technique

The basic idea of OPES [9] is to take as input a user-provided target distribution and transform the plaintext values in such a way that the transformation preserves the order.

OPES Works in Three Stages:

- **Model:** The input and target distribution are modeled as piece-wise linear splines.
- **Flatten:** The plaintext data is transformed into flat data.
- **Transform:** The flat data is transformed into the cipher such that the values in cipher data are distributed according to the target distribution.

b. Document Encoding and Encryption [5, 13, 14]

Using EPON value, create structural identifier for each node in tree. Then create integrity identifier using structural identifier and content at that node. Create encoding value for each node using structural identifier and integrity identifier. After encoding, apply encryption on encoded node using symmetric or asymmetric encryption technique. It is represented in figure 1. (b).

Using EPON value, create structural identifier for each node in tree. Then create integrity identifier using structural identifier and content at that node. Create encoding value for each node using structural identifier and integrity identifier. After encoding, apply encryption on encoded node using symmetric or asymmetric encryption technique.

Document encoding and encryption, introduce the notion of structural identifier for nodes in an XML document. This shows document encoding and an illustrative encryption technique.

i. Structural Identifier.
ii. Integrity Identifier.
iii. Document Encoding.
iv. Document Encryption.

c. Structure Based Routing [5, 13, 14]

In this, assign access control policies to role. Create content router which is basically is an application level router. Then setup the link between routers using document distance based routing technique. Publish content according to access control policy of role. Document verification process is depending on EPON value of each node in tree. In verification process, detect node dropping, structural integrity violation and integrity identifier. After verification process, create XML file using structural identifier of node and structural identifier of parent node. It is shown in figure 1. (c).

The Structure Based Routing involves the following entities [5, 13]:

- The document source is the document producer or a trusted owner of the document, and has full access to the original document and is the root of the multicast overlay network.
- The publisher publishes the data to a set of subscribers.
- The subscriber subscribes to the data and sends its request to a router-based publisher.
- The router routes the specific portion of the data to consumers and other routers.
- A router is both a publisher and a subscriber. The document source is a publisher. A consumer is a subscriber.
- A consumer is said to be associated with a router for a specific document if it has subscribed to that document through that router.

For simplicity of discussion, assume one document source. However the proposed solution can handle multiple document sources. A parent router of another router is one from which the latter receives some content. A child router is defined conversely. It uses multicast based approach to disseminate tree based data among consumers.

1. ***Access Permission:*** Access permissions on content for a user are expressed on a complete or partial sub tree of document.

2. ***Content Router:*** In Content Router, a router is an application level router that is able to route documents based on their structural organization which is represented by its structural identifier. Each router has an associated set of nodes in a document.

3. ***Dissemination Network:*** In document dissemination network, a link is between two content routers and intermediate network router developed using structural identifier. For development of dissemination network uses structural identifier. Subscription process is initiated by a consumer. Use a link setup protocol to establish a subscription link between two routers.

- **Document Distance Based Technique**

In this, Let e_x and e_z be EPON's of x and z nodes. e_z is PEPON at a router and e_x is EPON of the root of sub tree requested by another router. Then document distance calculated by $(e_x - e_z)$. After this is the router who has minimum document distance they selected as publisher router R_i. It defines how structural and EPONs can be used in establishing multicast paths.

- **Content Publishing**

If there is a nonempty set of routers that are subscribes for some nodes in document, then *R* forward document to these routers based on their requirements.

If R has a nonempty set of consumers for the document, it then forward document to consumer after encrypting it by using encryption technique but based on access control policies.

Content identification and extraction is carried out at each router that has at least one subscriber. Router keeps list of signatures of the root of sub trees that contains PEPON.

- ***Identification Step:*** Identification step determines the belongingness to relation among each of content roots accessible to each consumer and content sub trees it receives using simple EPON.

- ***Extraction Step:*** A depth first traversal is used to determine subscribed content root during extraction step. The EPON of root of subscribed content root is compared with EPON of visited node. If these EPONs match, then corresponding sub tree is extracted.

- **Document Verification [5, 13, 14]**

Document verification can be done at consumer side. The verification process uses the basic technique of tree traversal and hash computation. Therefore it is efficient and the implementation of such a technique is complex. The order of verification is linear in terms of the size of the content received because the post order traversal combined with the preorder processing on each sub tree verifies the integrity of the content.

- **Update Management:** Update to documents is either content or structure in the context of structure based routing.

  - ***Update Content:*** If there is any change in content then only the local hash of the node changes. The updates of the changed nodes along with their signatures are forwarded to the routers.

  - ***Update Structure:*** Structural changes have to be reflected in the mapping from user credentials to accessible nodes and their signatures. Therefore, the services that implement the mapping function from user credential to structural identifiers need to be notified accordingly with the new EPONs. If it is a distributed hash table, then the document source updates the hash table. The routers are also notified of the modifications. Removal of a sub tree is notified to the routers and consumers having the document.

  - ***Update Tree:*** In case of addition of a new sub tree, the original structure of the document is not affected. Therefore, the update is propagated to all the routers that have consumers with access permission to the new sub tree. In case of interchanges, the changes need to be propagated to the routers and consumers that are registered for any updated node or an ancestor of that updated node.

5. Attack Analysis

Basically there are two main attacks arises in dissemination of XML content, that could be conducted by the Publishers with the Owner [11, 12].

a. Subject Attacks
b. Publisher Attacks

a. **Subject Attacks:**

In this attack, subjects interact both with the Publishers and with the Owner. Subject implies that a malicious subject makes available his/her subject policy configuration to other subjects. Thus need to consider the possible attacks carried out in both interactions.

In the Owner-subject interaction, there exist two different kinds of attacks.

- **Authentication Attack.**
- **Passive Inference of Information.**

  - **Authentication Attack:**

  In an authentication attack, it implies the existence of a malicious subject S_m, which intrudes during the subscription phase of a subject S_i, the subject policy configuration. Thus, S_m can submit queries to the publishers exploiting the authorizations granted to S_i. The solution for this kind of attacks relies on the adoption of a standard authentication protocol during the subscription phase [11].

  In Authentication attacks, authorizations are revoked from a subject. In this case, the Owner regenerates the subject policy configuration, which is then sent to the subject. In such a scenario, it is necessary to avoid that a subject continues to use the old subject policy configuration, exploiting the revoked authorizations. For this purpose, assume that the Owner inserts the subject policy configuration, subject *ID* and the timestamp of the (re)generation. Then, the Owner stores in a public repository each subject *ID* and the timestamp of his/ her latest subject policy configuration. Moreover, when a subject is revoked, the Owner removes the relative entry from the public repository. In such a way, the Publisher can verify the validity of the subject policy configuration received by the subject.

- **Passive Inference of Information [11]:**

Another kind of subject attack is a passive inference of information from the secure structure. During completeness verification, the subject computes the hash values corresponding to the names of the nodes belonging to its authorized view. Then, by using such hash values, the subject is able to deduct the existence and the cardinality of these nodes in the secure structure. To avoid information inference, from the secure structure, it is possible to apply cryptographic techniques.

b. **Publisher Attacks:**

Elisa Bertino and, Bhavani Thuraisingham [11] shown that authenticity verification relies on the comparison between the Merkle hash values locally computed by a subject by using the information contained in the *MhPath* attributes, and the Merkle Signature contained in the Sign attribute, where all these attributes are contained into the Reply document [12]. Thus, a possible attack conducted by a Publisher is the replacement of the Sign element associated with an SE-XML document. In this case, even if the query result is authentic, the authenticity verification will always fail. To prevent this attack, we can associate with each XML document a unique *ID*, by inserting it as an attribute of its root. Thus, the Merkle hash value of the root of an XML document is computed on the hash value of its *ID*. Furthermore, the Owner must also sign, in addition to the Merkle hash value of the root of the XML document, the corresponding *ID* value. The resulting signature is then stored in the Sign element. The Publisher then inserts in the Reply document, together with the Sign element, also the attribute containing the corresponding *ID*. In such a way, the subject can verify at first the authenticity of the answer by using the Merkle hash value of the root. If the authentication process fails, he/she can verify if the *ID* value signed by the Owner matches the one in the Reply document and, thus, he/she can verify whether the above attack has occurred.

6. Result and Analysis

XML (Extensible Markup Language) is used as standard for document interchanges languages for the web.

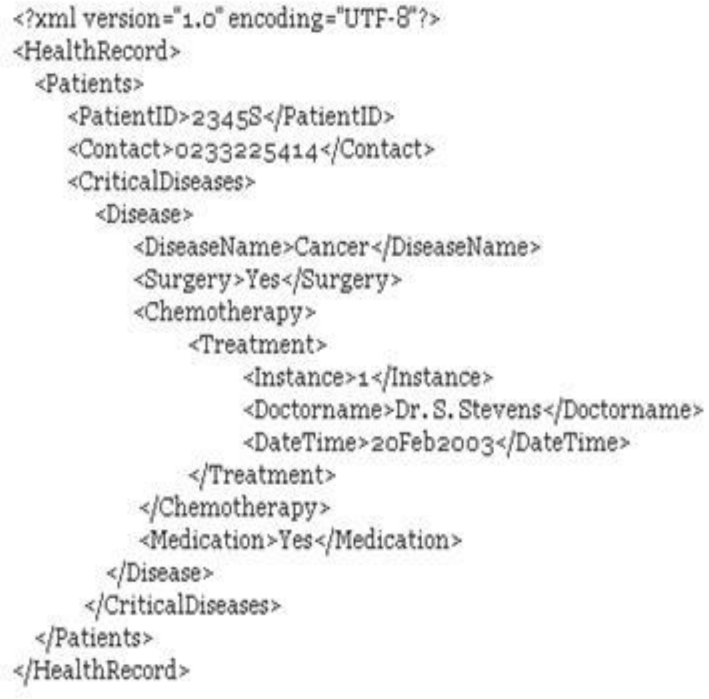
XML organizes data according to tree structure integrity and confidentiality of XML data is an important requirement for distributed web based application.

Merkle Hash Signature Scheme

The algorithms were executed on ***"HealthRecord.xml"*** file which shown in figure 2. Merkle proposed a digital signature scheme based on a secure conventional encryption function over a hierarchy (tree) of document fragments [11, 12]. Merkle trees are binding (integrity-preserving) but not hiding (confidentiality-preserving). The use of commutative hash operations to compute the Merkle hash signature prevents leakage related to the ordering among the siblings [11]. However it cannot prevent the leakage of signatures of a node and the structural relationships with its descendants or ancestors.

1. **Algorithm: Merkle Hash Technique.**

It takes input as a XML file then creates DOM tree representation for XML file. After the tree representation .it calculate hash tree using post order traversal technique and hash computation for each node. If node is attributing then calculate Merkle hash otherwise node is element. Further it calculate *sign'* using signature of subtree and concatenate with Merkle hash of each node M_hX_d. At the last Content and *Sign* send to the consumer along with the Hash values.



```
<?xml version="1.0" encoding="UTF-8"?>
<HealthRecord>
  <Patients>
     <PatientID>2345S</PatientID>
     <Contact>0233225414</Contact>
     <CriticalDiseases>
        <Disease>
           <DiseaseName>Cancer</DiseaseName>
           <Surgery>Yes</Surgery>
           <Chemotherapy>
               <Treatment>
                   <Instance>1</Instance>
                   <Doctorname>Dr. S. Stevens</Doctorname>
                   <DateTime>20Feb2003</DateTime>
               </Treatment>
           </Chemotherapy>
           <Medication>Yes</Medication>
        </Disease>
     </CriticalDiseases>
  </Patients>
</HealthRecord>
```

Figure 2. Input .XML (HealthRecord.xml) file

Merkle Hash Signature Verification Process

2. **Algorithm: In Merkle hash signature verification process.**

It takes input as XML Document + *Sign* and check whether it is verify or not. First it Generate *Sign''* using XML document, then compare *Sign''* with *Sign'*. If match then no integrity violation otherwise integrity for each node calculate Merkle hash value for each node in tree and compare with $M_hX_{d.}$ If all values are matched then no integrity violation occurred at that node.

Merkle proposed a digital signature scheme [5, 8, and 10] based on a secure conventional encryption function over a hierarchy (tree) of document fragments. The use of commutative hash operations to compute the Merkle hash signature prevents leakage related to the ordering among the sibling. In Merkle Hash Technique one way accumulation is very expensive in comparison to the one-way hash operation. Table I shows the Hash Value along with the node name.

Table 1. Hash Value Result Along With the Node Name Using Merkle Hash Signature Scheme

Node Name	Hash Value
PatientID	dbfaadb6b6f6690fe62b32d7d2bc44655dcafd29
Contact	9f0518e4cb499aee57265852ed0f3a878f018469
DiseaseName	3ec4da4707b08e492fa0f0070e2b422c26ecb306
Surgery	53767e8d2d91220d7f44237581adb43c1993d545

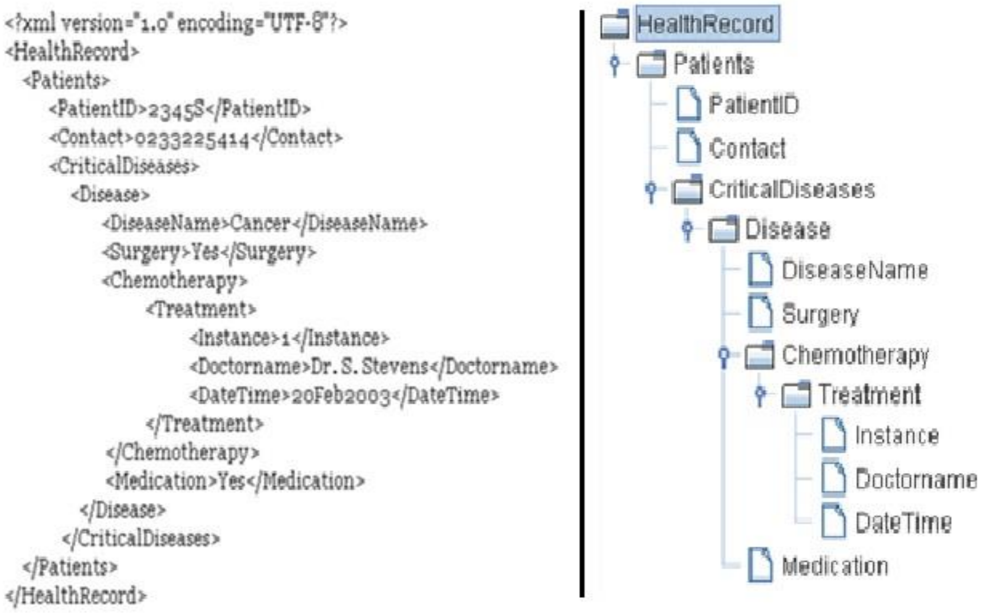
Instance	931545cea1f298e986d6f635ed75a7ee3e640464
Doctorname	e390092974a1d0210b6dbbe7e1c52e82d8f0925d
DateTime	c8ae3c5b9f3e611372e8dce6bbcf2d7a0fffa5da
Treatment	0113e227f58b82753c4834ce15bf0950098c1696
Chemotherapy	6dccc18e5154e3d05b27eda91143a9e443fbff10
Medication	1f3d18045ebd57705c5d226b469e82c21b000c0b
Disease	4feb07bd8b2ba583e8b0e08468b12b87349d7ca7
CriticalDiseases	86ca7eccadf1f8e05ef38fc95248f7a7e606c0de
Patients	055e775c1749379a9c47f4c4df4cca64451255cf
HealthRecord	acfbf2af1939ac8fba1324a2bacdd6519711a2cc
PatientID	dbfaadb6b6f6690fe62b32d7d2bc44655dcafd29

Drawbacks Merkle Hash Signature Scheme

1. It is binding (integrity) but not hiding (confidentiality), therefore it is vulnerable to inference attacks and data tampering attack.
2. If content of node changes then reconstruct hash tree.

Proposed Signature Scheme

Proposed Signature Scheme, take input as a XML (***HealthRecord.xml)*** file and create DOM tree representation for XML file shown in figure 3 (a) and 3 (b). Then assign post order number to each node in tree. Generate sorted random number and combine with post order number i.e. EPON numbers shown in Table II.



```
<?xml version="1.0" encoding="UTF-8"?>
<HealthRecord>
  <Patients>
     <PatientID>2345S</PatientID>
     <Contact>0233225414</Contact>
     <CriticalDiseases>
        <Disease>
           <DiseaseName>Cancer</DiseaseName>
           <Surgery>Yes</Surgery>
           <Chemotherapy>
                <Treatment>
                     <Instance>1</Instance>
                     <Doctorname>Dr. S. Stevens</Doctorname>
                     <DateTime>20Feb2003</DateTime>
                </Treatment>
           </Chemotherapy>
           <Medication>Yes</Medication>
        </Disease>
      </CriticalDiseases>
  </Patients>
</HealthRecord>
```

Figure 3. (a) *HealthRecord.xml* File (b) Hierarchical Tree Representation of *HealthRecord.xml* File

3. **Algorithm: Create Encrypted Post-order Number**

   First it Create a DOM representation of XML document. Then traverse the content tree in Post-order. Assign Post-order number *{*P_1,P_2, . . . ,P_n*}* for an XML data instance. Generate a random number r_x such that if node *y* is the last node visited prior to *x*, then $r_x \geq r_{y.}$ If all the nodes have been visited, then $\forall$ *x*, encrypt the combination of $P_{x.}$ and r_x using Order Preserving Encryption function.

These combined numbers are given as input to order preservation technique which creates encrypted post order number for tree.

4. **Algorithm: Order Preserving Encryption Function**

   Take input as bucket of $[p_l, p_h)$ with *h-l-1* sorted points $\{p_1, p_2, \ldots ,p_n\}$. *Then* set threshold for the bucket that is how many points in the bucket. To find linear spline for input bucket, for each point p_s in the bucket, compute its expected value. Afterwards split the bucket at the point that has the largest deviation from its expected value for each bucket. If the number of points in a bucket is below the threshold then stop splitting of the bucket otherwise find linear spline for input bucket. At last map calculated plaintext buckets into flat buckets using mapping function and scale factor of flat bucket. Then apply flat bucket into cipher bucket using mapping function and scale factor of cipher bucket Combine cipher bucket and got EPON values.

Table 2. Generation of Encrypted Post Order Numbering (EPON)

Node Name	PON (p_i)	EPON ($p_i + r_i$)
PatientID	1	14
Contact	2	22
DiseaseName	3	27
Surgery	4	39
Instance	5	43
Doctorname	6	49
DateTime	7	64
Treatment	8	66
Chemotherapy	9	79
Medication	10	86
Disease	11	93
CriticalDiseases	12	100
Patients	13	105
HealthRecord	14	114
PatientID	1	14

5. **Algorithm: Verification Process of Structural Signature Scheme**
   Take input as list of permitted subtree nodes L^u_p and reply for that permitted nodes $r^u{}_p$.

   Then check each value in $L^u{}_p$ With $r^u{}_{pi}$. If all values are matched then no dropping of node, else identified dropped node and stop. Decrypt each $r^u{}_{pi.}$ Suppose *x* is decrypted node. Then *x* has entries: $(S_x, S_{z,} <C_{x,} S_{z,}\ Content_x>$. Check value of S_z with inner S_z

and S_x with S_x in $C_{x.}$ If matches then no structural integrity violation, otherwise structural integrity violation occurred. Compute value of H_x using S_x and $Content_{x.}$ Compare H_x with I_x in $C_{x.}$ If matches then no content integrity violation, Else Content integrity violation occurred the stop. Using e_x of each node form XML document it verify or not.

Comparative Leaked Information during Verification Process result for Merkle Hash Signature Scheme and Proposed Signature Scheme shown in Table III.

Table 3. Leaked Information during Verification of Node using Merkle Hash Signature Scheme & Proposed Signature Scheme

Node	Nodes used	Leaked Information during verification of node using Merkle Hash Signature Scheme	Leaked Information during verification of node using Proposed Signature Scheme
PatientID	PatientID	None	None
Contact	Contact	None	None
DiseaseName	DiseaseName	PatientID, Contact, DiseaseName	None
Surgery	Surgery	None	None
Instance	Instance	None	None
Doctorname	Doctorname	Surgery, Instance, Doctorname	None
DateTime	DateTimedate	None	None
Treatment	Treatment	None	None
Chemotherapy	Chemotherapy	DateTimedate, Treatment, Medication	None
Medication	Medication	None	None
Disease	Disease	None	None
CriticalDiseases	CriticalDiseases	None	None
Patients	Patients	Disease, CriticalDiseases, Patients	None
HealthRecord	HealthRecord	None	None
PatientID	PatientID	None	None

Table 4. Time Required For Verification Process For Merkle Hash Signature Scheme & Proposed Signature Scheme In Sec. For Different Of Nodes

Number Of Nodes	Time for Verification Process For Merkle Hash Signature Scheme in Sec.	Time for Verification Process for of Proposed Signature Scheme in Sec.
4000	0.422	0.463
8000	0.623	0.782
12000	0.82	0.909
16000	1.201	1.509
20000	2.01	2.343

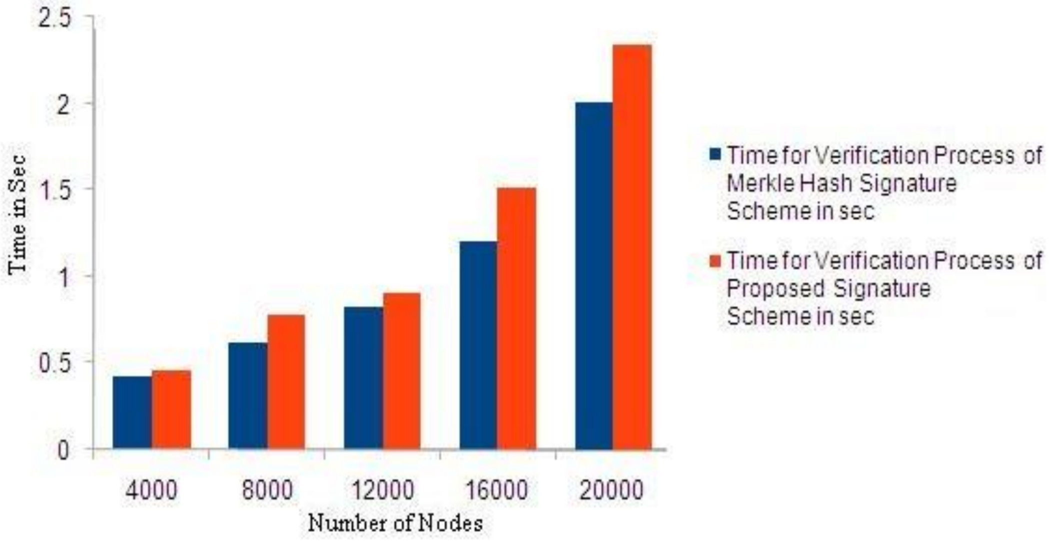


Figure 4. Graphical Representation of Performance Analysis of Time Required for Verification Process for Merkle Hash Signature Scheme & Proposed Signature Scheme in Sec. for different of Nodes

Table III shows comparison of result leaked information during verification of node in both techniques. Merkle hash technique is sensitive to inference attack and data tampering attack. Merkle has trees are binding (integrity-preserving) but not hiding (confidentiality-preserving) the information. The implemented algorithms are tested for different number of nodes, with respective to time. TABLE IV shows the time required for verification process for Merkle Hash & Proposed Signature Generation scheme with respect to its number of nodes. If numbers of nodes are increases then time required for Proposed Signature Scheme and Merkle Hash Signature Scheme is also increase. Figure 4, shows the graphical representation of performance analysis of Merkle Hash & Proposed Signature Scheme.

7. Conclusion

As it is very often case in most of the organizations, whenever large amounts of data to be shared among users, one must put in place suitable access control policies. Once access control policies are stated, they are implemented by an access control mechanism. Because XML is becoming the most prevalent means according to which documents and data are encoded for distribution among users on the Web, there is a strong need for models and mechanisms enabling the specification and enforcement of access control policies for XML documents. Such models and mechanisms are crucial in order to facilitate a selective dissemination of XML documents, containing information of different sensitivity levels, among user communities.

Proposed Signature Scheme solves problem of post order numbering using order preserving encryption technique. In Order preserving encryption scheme, comparison, equality queries can be directly applied to encrypted data and does not require any decryption process. Role based access control policy depend on EPON numbering which ensures that a consumer is delivering only the portion of data that it has access. Result of query processing over data encrypted using EPON are exact. Proposed Signature Scheme provides stronger security in terms of integrity and confidentiality. It simplifies the transmission of tree based data from a publisher to consumer and improves efficiency of such transmission. Flexibility in security enforcement is known to be an important requirement in secure system design and implementation. Depending on the degree of trust on the network integrity, checks may or may not be enforced. Moreover, this Proposed Signature Scheme facilitates dissemination of contents with varying degrees of confidentiality and integrity in a mix of trusted and untrusted networks, which is so prevalent in current settings across enterprise networks and the web.

References

[1] M. Altinel and M. J. Franklin, "Efficient Filtering of XML Documents for Selective Dissemination of Information", Proceedings of VLDB Conference, **(2000)**.

[2] A. Crespo, O. Buyukkokten, and H. Garcia-Molina, "Query merging: Improving query subscription processing in a multicast environment", IEEE Trans. Knowl. Data Eng., vol. 15, no. 1, **(2003)**, pp. 174-191.

[3] “Document Object Model (DOM)” [Online], Available: http://www.w3.org/DOM.

[4] E. Bertino, B. Carminati, E. Ferrari, B. M. Thuraisingham and A. Gupta, “Selective and Authentic Third - Party Distribution of XML Documents", IEEE Trans. Knowl. Data Eng., vol. 16, no. 10, **(2004)**, pp. 1263-1278.

[5] A. Kundu and E. Bertino, "A New Model for Secure Dissemination of XML Content", IEEE Transaction on System and Cybernetics (Part C: Application and Reviews), vol. 38, no. 3, **(2008)**.

[6] Extensible Markup Language (XML) [Online], Available at http://www.w3.org/XML/.

[7] R. Agrawal, J. Kiernan and R. Srikant, “Order Preserving Encryption for numeric data", Proc. ACM SIGMOD Int. Conf. Mana. Data, **(2004)**.

[8] V. N. Waghmare and R. C. Thool, “Implementation of Efficient Signature Scheme for Leakage Free Dissemination of XML Content using Structure Based Routing”, Computer Engineering, Elixir Comp. Engg., vol. 81, **(2015)**, pp. 31493-31498.

[9] E. Bertino and E. Ferrari, “Secure and Selective Dissemination of XML Document", ACM Trans. Inf. Syst. Secur., vol. 5, no.3, **(2002)**, pp. 290-331.

[10] V. Waghamre and R. Thool, “A Review on: Issues Related to Security on Tree Structure Data”, IJCSI International Journal of Computer Science Issues, vol. 10, no. 2, **(2013)**, pp. 210-214.

[11] E. Bertino and B. Thuraisingham, “Selective and Authentic Third-Party Distribution of XML Documents”, IEEE Transactions on Knowledge and Data Engineering, vol. 16, no. 10, **(2004)**, pp. 1263-1278.

[12] W. Stallings, “Network Security Essentials: Applications and Standers”, **(2000).**

[13] Kundu and E. Bertino “Secure dissemination of XML content using structure based routing", Proc. 10th IEEE Int. Enterprise Distrib. Object Computer. Conf. (EDOC)", **(2006)**, pp. 153-164.

[14] V. Waghamre and R. Thool, “Encrypted Signature Scheme for Secure & Selective Dissemination of Tree Structured Data (XML)”, IASET’s International Journal of Computer Science and Engineering (IJCSE), vol. 3, no. 3, **(2014)**, pp. 31- 40.

[15] N. Nagy and M. E. El Sharkawi, “XML Security: Detecting and Preventing Disclosure in XML”, Journal of Information Assurance and Security, vol 3, **(2008)**, pp. 212-219.

[16] A. Kundu and E. Bertino, “Structural Signatures for Tree Data Structures”, VLDB Endowment, ACM, **(2008)**.

[17] P. Devanbu, M. Gertz, C. Martel and S. Stubblebine, “Authentic Third Party Data Publication", Proc. 14th Ann. IFIP WG 11.3 Working Conf. Database Security, **(2000)**.

[18] Daconta and Sangich, “XML Development with Java2”.

[19] D. Hook, “Beginning Cryptography with Java”.

[20] A. Chan, "Transactional Publish/Subscribe: The Proactive Multicast of Database Changes", ACM SIGMOD, **(1998)**.

[21] M. Tan, M. D. Theys, H. J. Siegel, N. B. Beck and M. Jurczyk, “A Mathematical Model, Heuristic and Simulation Study for a Basic Data Staging Problem in a Heterogeneous Networking Environment", Proceedings of the 7th International Computing Workshop (HCW), IEEE, **(1998)**.

[22] Q. Hu, D. L. Lee and W. C. Lee, "Optimal Channel Allocation for Data Dissemination in Mobile Computing Environments", 18th International Conference on Distributed Computing Systems, **(1998)**.

[23] M. Lazaro and P. Sage, “Any Information, Anywhere, Anytime for the Warghter", Proceedings of the SPIE, vol. 3080, **(1997)**, pp. 35-42.

Authors

Vivek N. Waghmare, he completed his B. Tech., M. Tech. from SGGS Institute of Engineering & Technology, Nanded, and Walchand College of Engineering, Sangli, India in 2008 and 2010 respectively. He is currently pursuing his Ph.D. at SGGS Institute of Engineering & Technology, Nanded under SRTMUN, Nanded, India. His research area includes HPC, Network Security.

Ravindra C. Thool, he received his BE, ME and Ph.D. in Electronics from SGGS Institute of Engineering & Technology, Nanded, India, in 1986, 1991 and 2003 respectively. He is currently working as professor and head with Information Technology department in the same organization. His research area includes Computer Vision, Image processing and multimedia information systems. He has published several research papers in refereed journals and professional conference proceedings. He is member of IEEE, Life member of ISTE.