International Journal of Computer Science
and Engineering (IJCSE)
ISSN(P): 2278-9960; ISSN(E): 2278-9979
Vol. 3, Issue 3, May 2014, 31-40
© IASET

International Academy of Science,
Engineering and Technology
Connecting Researchers; Nurturing Innovations
IASET

ENCRYPTED SIGNATURE SCHEME FOR SECURE & SELECTIVE DISSEMINATION OF TREE STRUCTURED DATA (XML)

VIVEK N. WAGHMARE[1] & RAVINDRA C. THOOL[2]

[1]Research Scholar, S.G.G.S.I.E & T, S.R.T.M University, Nanded, Maharashtra, India

[2]Professor & Head (IT), S.G.G.S.I.E & T, S.R.T.M University, Nanded, Maharashtra, India

ABSTRACT

Most of the data represent on web is hierarchical data structure. The main problem of tree structured data is that, to secure and making it available in efficient manner both in internal & external level. Dissemination of data address the issues if verified integrity of data without any leakage and selective and secure distribution of data in network

In proposed approach, Encrypted Signature Scheme binds as well as hides the information. Encrypted Post Order Numbering (EPON) overcomes the vulnerabilities of Post Order Numbering (PON). It is based on the structure of tree, as defined by post order tree traversal technique. It uses a randomized notion and Order Preserving Encryption Scheme (OPES) of such traversal numbers. This technique not only prevents the information leakage, but also provides better security for tree structured data.

KEYWORDS: Confidentiality, Epon, Integrity, Pon, Opes, Vulnerability

INTRODUCTION

An XML (Extensible Markup Language) has become the standard document for inter change language on the web. XML document contain information of different sensitivity degrees that must be shared by possible large number of user. Data sharing among multiple parties require both data integrity & data confidentiality [9]. Data that a consumer is not authorized to access, but belongs to the complete data set is called extraneous data. Sending of extraneous data to a consumer may leak in information, even if the data is encrypted. In general, the extraneous data is prone to off-line dictionary attacks even by a legitimate consumer that can exploit contextual knowledge from the data elements it has access to. Therefore, it is most important to remove, the extraneous data, even if the contents are encrypted with keys that the consumer does not have, before its delivery.

Efficiency and scalability must however be provided by assuring at the same time security of contents and privacy of the parties acquiring and disseminating contents [3]. It is not useful to provide high-bandwidth content distribution systems if the integrity of disseminated contents are not as sure or the property of the contents not protected [11]. Such problems are further complicated when dealing with contents encoded in XML, because of the hierarchical organization of the contents of tree structure data, different confidentiality as well as integrity requirements may exist for different portions of the same content [4] [14]. Thus, which leads to need of dissemination approach to XML that, addresses the issues of security, privacy and scalability in a holistic manner [2]. The structural properties are also contributed towards the efficient and scalable dissemination framework. This solution is based on the simple notion of encrypted Post Order Numbering [1] and its properties. A key feature of this approach is that it directly takes into account access control policies, specifying which entity can access which portion of the contents, so that contents is disseminated according to their policies. Dissemination of content in wide area network is an important concern of publish-subscribe system especially for hierarchical data model. Hierarchical data model require different security for different portion of da**ta. Dissemination of** data address the issues if verifies integrity of data without any leakage and selective and secure distribution of data in network [20].

Requirement for Such a Dissemination Approach Include the Following

Access Control: To prevent unauthorized users to infer sensitive information, they are authorized to access.

Data Integrity: Not only the integrity of the data must be verifiable by the user, but also any compromise to the data must be precisely determined.

Data Confidentiality: A user receives only that information that user is allowed to access, by the defined access control policies and not able to infer any sensitive information that user is not authorized to access.

RELATED WORK

Many of different design approaches are related to this topic. In this first area, Bertino and Ferrari approach supporting access control in both pull and push based distribution of data [3]. This approach is depending on encrypting different portion of the data with different keys & then distributing the keys to data consumers according to access control policies. Information pull is based on authorization. Consumer sends the request to source for XML document. When consumer submits an access request then [15] 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, the access is denied. Information push approach is used for distributing documents to users which based on broadcast data to clients. Also 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 [5]. Example, the case of a newsletter sent once a week to all users. Different users have different privilege to see 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 [20].

The main problem with Information pull and Information push approach is number of views becomes large and such approach cannot be practically applied. Bertino [4] have also investigated the problem of integrity of XML data by using notion of Merkle Hash Tree. Merkle proposed a digital signature scheme based on a secure conventional encryption function over a hierarchy (tree) of document fragments. Merkle trees are binding (integrity-preserving) but not hiding (confidentiality-preserving) [12]. The use of commutative hash operations to compute the Merkle hash signature prevents leakage related to the ordering among the siblings. However it cannot prevent the leakage of signatures of a node and the structural relationships with its descendants or ancestors. Moreover, one way accumulation is very expensive in comparison to the one-way hash operation. The Merkle hash technique has been widely used in data authentication.

Drawback of Merkle Hash Tree Technique,

- It is not scalable.
- It does not remove extraneous data from contents.

- It is binding (integrity) but not hiding (confidentiality).
- It vulnerable to inference attacks and data tampering attack.
- In Markle Hash Tree Technique, if content of node changes then reconstruct the hash tree.

IMPLEMENTATION DEATILS

XML (Extensible Markup Language) [9] is a used as standard for document interchanges languages for the web. It is platform for application integration and management on the Internet. XML is used in critical areas such as government, finance, healthcare and law [20]. XML document contain information of different sensitivity degrees that must be shared by possible large user communities.

Dissemination of XML content addresses the issues of security, privacy. Building blocks of XML documents are nested, tagged elements. Each tagged elements has zero or more elements, zero or more attributes, and may contain textual information that is data content. Elements can be nested at any depth in the document structure [9]. 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 then *T* be a DOM [10] tree representation of *D*. *T (V, E)* where *V* be a set of vertices and *E* be a set of edges. *T* is a nonlinear, acyclic data structure. $Content_x$ be content only at *x*. $Content_x$ contains only the content specific to *x* and not of other nodes [20].

Post Order Numbers (PON) [6] [7] [20]

In this, take input as a XML file and create DOM tree for XML file. Then assign post order number to each node in tree. Let p_x be Post Order Number assigned to each node in tree according to Post Order traversal of tree [6]. 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*.

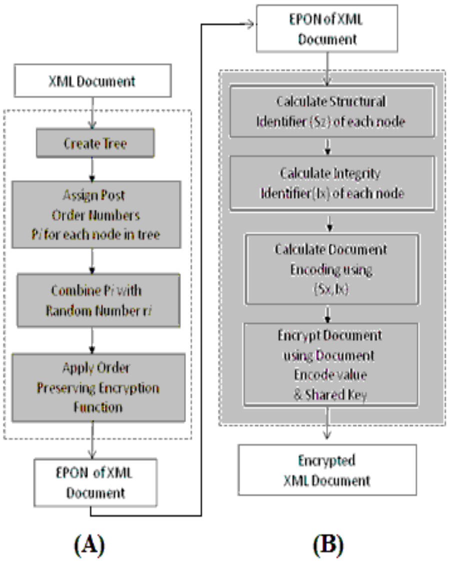

XML Document
Create Tree
Assign Post Order Numbers Pi for each node in tree
Combine Pi with Random Number ri
Apply Order Preserving Encryption Function
EPON of XML Document
EPON of XML Document
Calculate Structural Identifier (Sz) of each node
Calculate Integrity Identifier(Ix) of each node
Calculate Document Encoding using (Sx,Ix)
Encrypt Document using Document Encode value & Shared Key
Encrypted XML Document
(A)
(B)

Figure 1: (A) Block Diagram for Generation of Encrypted Post Order Number (B) Block Diagram of Encoding and Encryption of Document

GENERATION OF ENCRYPTED POST ODER NUMBER [6] [7] [20]

EPON [1] [2] generation module overcome security related flaws of solutions based on the use of PON. 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 block diagram 1A.

Algorithm: Create Encrypted Post-order Number.

Input: XML Document.

Output: Encrypted Post Order Number for each node.

- Create a DOM representation of XML document.
- 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.

Computation

Let $\{p_1, p_2,...., p_n\}$ be a set of PONs for an XML document. Each p_i, i = 1, 2, …, n is combined with a unique random number *ri*. 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. Let, *x, y,* and *z* be nodes such that *x* and *y* are children of *z.* let p_x, p_y, and p_z be their PONs, respectively. The random values would be r_x, r_y, and r_z, respectively. By definition of PONs, $p_z > p_x$ and $p_z > p_y$. $r_z > r_x$ and $r_z > r_y$, that is, the order of the PONs is preserved by the random numbers. r_x and r_y should be chosen so that no relation can be and should possibly be established between them.

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 [5].

Algorithm: Order Preserving Encryption Function:

Input: Plaintext value that is combined value of P_x and $r_{x.}$

Output: Cipher value that is Encrypted Post Order Number.

- Take input as bucket of $[p_l, p_h)$ with *h-l-1* sorted points $\{p_1, p_2,..., p_n\}$
- Set threshold for the bucket that is how many points in the bucket

- Find linear spline for input bucket,
- For each point p_s in the bucket, compute its expected value
- 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.
  - Else go to step 3.
- 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.

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.

Document Encoding and Encryption [6] [7]

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 [7]. After encoding, apply encryption on encoded node using symmetric or asymmetric encryption technique [6] [7], as shown in block diagram 1B.

- **Structural Identifier**

Let *z* be a node and *Sz* be a structural identifier of node z, is defined as: $S_z = (e_z, e^z{}_{lowest})$.

Where, e_z is EPON value of *z* and $e^z{}_{lowest}$ is the lowest EPON for any node in the set of descendant nodes of *z*. The structural identifier is unique for each node in a document element and also identifies subtree of *z*.

- **Integrity Identifier**

The content of a node includes attributes of an XML element, but does not include any of its descendents. Integrity identifier of node x is defined as: $I_x = H\ (S_x, Content_x)$.

Where, $Content_x$ is content at node *x*, *H* is a one way collision-resistant hash function.

- **Document Encoding**

Each node *x* in a document has encoding information C_x defined as tuple: $C_x = (S_x, I_x)$.

If each node *x* with parent *z* in a content tree is encoded with tuple $<C_x, S_z>$ else if *x* is the root then its encoding is $<C_x>$.

- **Document Encryption**

Document encryption is applied on document encoding. Each encoded node is encrypted using a key that is shared between producer and consumer.

Document Encryption E^{x}_{s} of node x is: $E^{x}_{s} = K_{s} (E^{x}_{m})$, Where, $E^{x}_{m} = (C_{x}, S_{z,} x)$ and K_{s} is Shared Key.

After encryption, each document node x is represented as: $<S_{x}, S_{z,} E^{x}_{s}>$, Where S_{z} is structural identifier of each node z, parent of x.

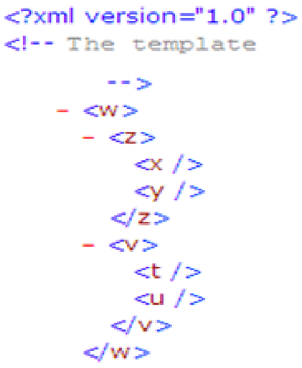
Document Verification

Document verification can be done at consumer side. The following steps must be executed for document verification [6] [7]

- If nodes have been dropped.
- If the order of the nodes has been changed.
- If the content of a node has been compromised.
- If some nodes have been added in an unauthorized manner.
- If the content of one node has been replaced with the content of another node.
- Document verification can be done at consumer side.

RESULT 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. The XML document is shown in Figure 2.



```
<?xml version="1.0" ?>
<!-- The template
     -->
  - <w>
    - <z>
        <x />
        <y />
      </z>
    - <v>
        <t />
        <u />
      </v>
    </w>
```

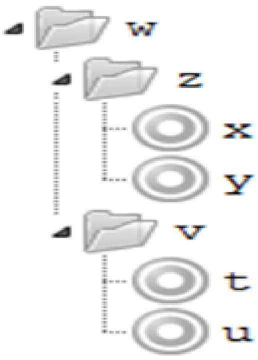


Figure 2: XML FILE **Figure 3: Hierarchical Tree Representation of.XML File**

Figure 3. Shows the hierarchical tree representation of.XML. Document Object Model (DOM) tree of.XML shows in Figure 4. The DOM is an object-oriented interface for accessing XML documents that have been parsed into an object-oriented representation of the XML document tree. After assigning post order number to each node in tree. Generate sorted random number and combine with post order number as shown in Figure 5. Figure 6, shows EPONs for the tree generated by using PON as that shown in Figure 5. The verification process is efficient and simple that can be done at consumer by using shred key to decrypt the encrypted content of xml file. It uses the basic technique of post and pre order

traversal and hash computation. Therefore the computation is not expensive nor is the implementation of such a technique 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. EPONs preserve the order of PONs; therefore properties characterizing EPONs are identical to those of PONs. HX is I_x, EX is e_x, EXLOW is $e^x{}_{lowest,}$ S_Z is e_z and S_{ZLOW} is $e^z{}_{lowest}$.

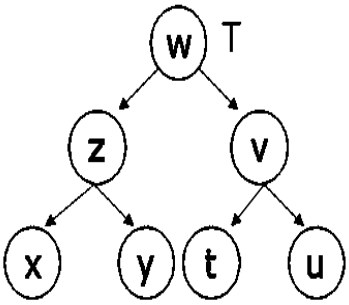


Figure 4: DOM Tree Representation of.XML File

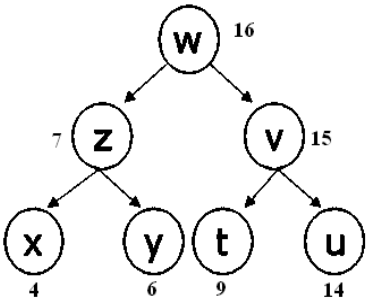


Figure 5: Generation of PON Values to Tree

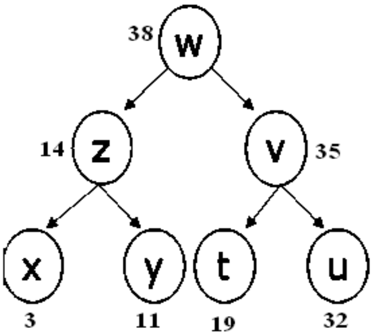


Figure 6: Generation of Epon Values to Tree

Table 1 shows comparison result for leaked information during verification of node in Merkle Hash & Encrypted Signature Generation Scheme. From Table 1, it is observed that, Merkle Hash technique binds but does not hide the information, while Encrypted Signature Scheme binds as well as hides the information. However, Merkle hash technique cannot prevent the leakage of signature of nodes and the structural relationship with its descendent & ancestors.

Table 1: Verification Process of Using Merkle Hash Technique & Encrypted Signature Scheme

Nodes	Nodes Used	Leaked Information During Verification of Node in Merkle Hash Signature Scheme	Leaked Information During Verification of Node in Encrypted Signature Scheme
X	X	None	Define upper and lower bounds of subtree
Y	Y	None	“
Z	X, Y, Z	Signature of x, y; y as sibling of x and x, y are child of z	“
T	T	None	“
U	U	None	“
V	T, U, V	Signature of t, u; u as sibling of t and t, u are child of v	“
W	Z, V, W	Signature of z, v; v as sibling of z and v, z are child of w	“

Table 2: Time Required for Generation of Merkle Hash & Encrypted Signature Scheme

Number of Nodes	Time for Generation of Merkle Hash Signature Scheme in Sec	Time for Generation of Encrypted Signature Scheme in Sec
1000	0.230	0.172
2000	0.451	0.258
3000	0.551	0.318
4000	0.725	0.435
5000	0.981	0.579

The results are captured and analyzed, for both the algorithms, i.e. Merkle Hash & Encrypted Signature Schemes. The implemented algorithms are tested for different number of nodes, with respective time. Table 2. shows that, as the number of nodes increases the time required to generate, the respective Signature Schemes are also increases.

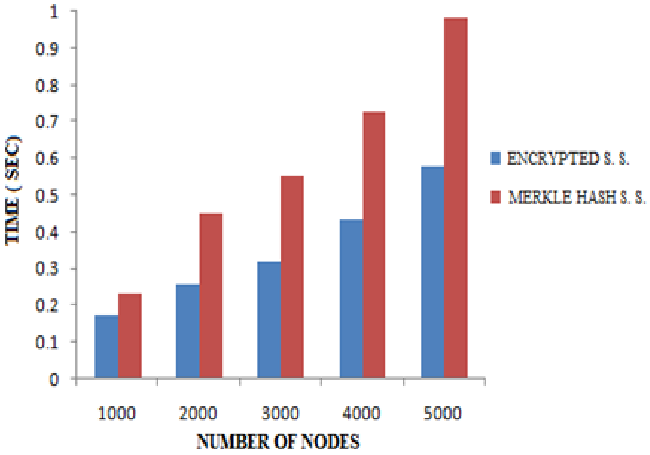


Figure 7: Graphical Representation of Performance Analysis of Merkle Hash & Encrypted Signature Generation Scheme

Figure 7, Shows the graphical representation of performance analysis for Merkle Hash & Encrypted Signature Schemes. Encrypted Signature Generation Scheme requires less time and imparts better security than Merkle Hash technique.

CONCLUSIONS

Encrypted Post-Order Numbering, used for verify integrity of data for secure trasmission. Using access control policies transfer of only selected data to the authorized user is possible. Encrypted Signature Scheme, simplifies the transmission of XML data from a publisher to consumer and improves efficiency of such transmission. This technique avoids sending the extraneous data that are not accessible to consumer and protects against data tampering attack and inference attack. Thus, this approach represents an efficient and secure mechanism for use in applications such as publish-subscribe systems for XML documents.

REFERENCES

1. M. Altinel and M. J. Franklin. "Efficient filtering of XML documents for selective dissemination of information", in Proc. VLDB Conf., 2000, pp. 53-64.

2. A.Crespo, O. Buyukkokten, and H. Gracia-Molina, "Query merging: Improving query subscription processing in a multicast environment", IEEE Trans. Knowl. Data Eng., Vol. 15, no.1, pp. 174-191, 2003.

3. E. Bertino and E. Ferrari, "Secure and selective dissemination of XML documents", ACM Trans. Inf. Syst. Secur., vol. 5, no. 3, pp. 290331, 2002.

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, pp. 12631278, Oct.2004.

5. R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu, "Order preserving encryption for numeric data", in Proc. 2004 ACM SIGMOD Int. Conf. Mana.Data, pp. 563574.

6. Kundu and E. Bertino, "Secure dissemination of XML content using structure based routing", in Proc. 10th IEEE Int. EnterpriseDistrib. Object Comput. Conf. (EDOC06), 2006, pp. 153164.

7. Kundu and E. Bertino, "A new model for Secure Dissemination of XML Content", IEEE Transaction on, May 2008.

8. G. BAnvar, T. Chandra, B. Mukharjee, "An efficient multi-cast protocol for content based publish-subscribe systems", ACM Symp., GA, 1999.

9. Extensible Markup Language [Online]. Available: http://www.w3.org/XML/.

10. Document Object Model (DOM) [Online]. Available: http://www.w3.org/DOM/.

11. K. Datta, M. Gradinariu, M. Raynal, and G. Simon, "Anonymous publish subscribe in p2p networks", presented at the Int. Parallel Distrib.Process. Symp., Nice, France, 2003.

12. F. Cao and J. Singh, "Efficient event routing in content-based publish subscribe service Networks," in Proc. of IEEE INFOCOM 2004, pp. 929-940.

13. Q. Hu, D. L. Lee, and W. C. Lee, “Optimal Channel Allocation for Data Dissemination in Mobile Computing Environments”, In 18th International Conference on Distributed Computing Systems, May 1998.

14. M. Lazaro and P. Sage, “Any Information, Anywhere, Anytime for the Warghte”, In Proceedings of the SPIE, volume 3080, pages 35-42, 1997.

15. 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, Aug.2000.

16. FERNANDEZ, E., GUDES, E., AND SONG, H. "A model for evaluation and administration of security in object-oriented databases", IEEE Trans. Knowl. Data Eng, 1994, 275292.

17. R. Douglass, J. Mork, and B. Suresh. Battleeld “Awareness and Data Dissemination (badd) for the Warghter,” In Proceedings of the SPIE, volume 3080, pages 18-24, 1997.

18. R. Lindell, J. Bannister, C. DeMatteis, M. O'Brien, J. Stepanek, M. Campbell, and F. Bauer. “Deploying Internet Services Over a Direct Broadcast Satellite Network: Challenges and Opportunities in the Global Broadcast Service,” In MILCOM. IEEE, 1997.

19. T. Stephenson, B. DeCleene, G. Speckert, and H. Voorhees. Badd phase ii. Dds “Information Management Architecture,” In Proceedings of the SPIE, volume 3080, pages 49-58, 1997.

20. Vivek N. Waghmare, Dr. Ravindra C. Thool, ”A Review on: Issues Related to Security on Tree Structure Data”, In IJSCI International Journals of Computer Science Issues, Vol. 10, Issue 1, No 2, 210-214, January 2013.

AUTHOR’S DETAILS

Vivek N. Waghmare, 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.

Dr. Ravindra C. Thool, 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.