Journal of Computer Science, 9 (4): 427-432, 2013
ISSN 1549-3636

doi:10.3844/jcssp.2013.427.432 Published Online 9 (4) 2013 (http://www.thescipub.com/jcs.toc)

Temporal, Delegable and Cheap Update Access Control to Published XML Documents

[1]Waleed Halboob, [2]Ali Mamat, [2]Ramlan Mahmod and [1]Muhammad Khurram Khan

[1]Center of Excellence in Information Assurance,
King Saud University, Riyadh, Saudi Arabia
[2]Faculty of Computer Science and Information Technology,
Universiti Putra Malaysia, 43400 Serdang, Selangor, Malaysia

Received 2012-05-02, Revised 2012-12-17; Accepted 2013-05-08

ABSTRACT

Providing access control for published XML documents on the Web is an important topic. It involves the use of cryptographic techniques, addressing different requirements and, as a result, facing several challenges. Existing solutions still have some weaknesses such as system update cost, number of required secret encryption/decryption keys, size of encrypted document and supporting temporal and delegable access. This study propose a push--based access control policy enforcement mechanism for addressing these issues using a Dynamic Key Management Table (DKMT) and based on Identity Based Encryption (IBE). The proposed mechanism addresses the existing challenges and provides a more acceptable solution.

Keywords: XML Documents, XML Access Control, Access Delegation and Temporal Access, Secure Socket Layer (SSL), IP Security (IPsec), Dynamic Key Management Table (DKMT)

1. INTRODUCTION

The XML language has become a de facto standard for data exchanging and transmitting on the Web. XML documents disseminated on the Web (Chen, 2007). XML document may contain secret data that must be protected from unauthorized use so XML access control is an important topic in Web information security (Bertino *et al*., 2001). A comprehensive XML access control system includes two parts, policy specification and policy enforcement (Bertino *et al*., 1999). The policy specification is a policy language used for specifying the access control policies while the policy enforcement is a mechanism used for enforcing the specified policies during the delivery of the document to users.

Based on the policy enforcement, the existing XML access control systems are classified into two categories pull-based and push-based systems (Bertino *et al*., 1999; Miklau and Suciu, 2003). In the pull-based systems, a server receives a request from a user and responds with a proper result (document or document portion). Because the view is sent to only one user, it can be protected using the existing network security techniques such as IP security (IPsec) or Secure Socket Layer (SSL) protocols. In the push-based systems, the server periodically publishes or broadcasts the whole document to all users. Hence, the document cannot be completely protected using IPSec or SSL protocols since the document's nodes have different access by different users. As a solution, the nodes are encrypted with different secret keys which are then distributed to users in a way that each user receives only the secret keys of the nodes that they are authorized to access.

Both pull-based and push-based system must address some requirements such as supporting a fine-grained access and XML scheme (DTD or XML Scheme). Although, the policy enforcement in push-based systems faces new challenges such as (Bertino *et al*., 1999; 2001; 2003; Zhang *et al*., 2005; Crampton, 2004):

- Encrypting the document's nodes requires the use of a large number of secret keys basically equal to the

Corresponding Author: Waleed Halboob, Center of Excellence in Information Assurance, King Saud University, Riyadh, Saudi Arabia

document's nodes as these nodes have different access by different users. Furthermore, for each document, the user has to manage several secret decryption keys also basically equal to his/her related nodes
- For each published document, a secure channel is required for distributing the secret keys to each user. This increases the system overhead as the right secret keys must be selected for each user and then distributed over a secure channel
- The size of the user's decrypted document is large since it contains unneeded encrypted nodes that the user is not authorized to access. This affacts the user's storage capacity which is not likely to be as large as the server storage
- System update (user joins/leaves) are expensive tasks. For example, if the user leaves, the system must change the secret keys that this user had obtained and distribute new secret keys to all of the affected users
- How to support user with a temporal access upon his subscription period
- Providing user with delegable access-as a service-to enable him to delegate his access to other users

Push-based dissemination is now widely used by many service providers. It allows users to easily receive data anywhere and in an efficient way. Several organizations deliver their data using a push-based dissemination mode. For example, pay-per-view commercial channels selectively publish their video and audio data to subscribed users. Another example is digital libraries which distribute their new issues (e.g., newspapers). For that reason, addressing the challenges mentioned above is a non-trivial issue. These challenges have been investigated by several models but still have weaknesses. Moreover, reducing the size of a user's decrypted document and providing a user with delegable access still research gaps.

In this study, we propose a policy enforcement mechanism to address the above challenges using a Dynamic Key Management Table (DKMT) and based on the Identity Based Encryption (IBE). Our proposed mechanism relies on our trust-based XML policy language and push-based XML access control model proposed in (Halboob *et al*., 2008).

The rest of this study starts with a discussion of related works followed by presenting the used matrials and methods and finally the discussion and result of our proposed work followed by conclusion.

1.1. Related Works

Several push-based access control models have been proposed for addressing the challenges discussed above. The Author-X system (Bertino *et al*., 2001) is the first effort which reduced the number of required secret encryption/decryption keys to less than the number of nodes. But, this scenario is not efficient because it involves a system with long matching and marking processes that must be repeated if any update happened. For each published document, a secure channel is required with each user to distribute his secret decryption keys. The Author-X system was extended in (Bertino *et al*., 2003; 2007) to support temporal access by using temporal policies and keys. Using temporal policies requires replicating a policy several times if it is assigned to several users with different access periods, which affects the system scalability. Miklau and Suciu (2003) proposed a framework that uses three secret keys for encrypting each node and based on super-encryption concept, in which the lower level nodes are first encrypted, then the higher level nodes and so on. Although super-encryption has an advantage in term of producing a well-formed encrypted document, it has three disadvantages which are: (i) it needs a huge number of secret encryption keys (three for each node); (ii) it does not seem efficient as each node is encrypted several times depending on its level; and (iii) it produces a large encrypted XML document in terms of size because a metadata is added to the encrypted nodes. Crampton (2004; 2006) proposed another model that also relies on the super-encryption concept used in (Miklau and Suciu, 2003). This model reduces the number of secret encryption keys to the number of nodes and gives the user only one secret decryption master key, but this key is not permanent, which means it is used with one published document only. By extending the Author-X system (Bertino *et al*., 2001; Zhang *et al*., 2004; 2005) proposed a model that uses a key distribution scheme proposed in (Mu and Varadharajan, 2001) and gives each user only one private permanent decryption key and reduces the frequency of the secure channel used to only once for each user during his access period. This model also reduces the system update cost but only with a limited group of users. Bouganim *et al*. (2008) and Ko *et al*. (2007) investigate the efficiency of the policy enforcement without considering the above challenges.

Table 1 summarizes the results achieved by related works based on the challenges presented earlly. These results show that there is a need for reducing the number of secret keys, size of the user decrypted document, system update cost while supporting user with a temporal and delegable access.

Table 1. Related works' achievements

Challenges	Author-X system (Bertino *et al*., 2001; 2003; 2007)	Miklau and Suciu (2003) model	Crampton (2004; 2006) model	Zhang *et al*. (2004; 2005) model
No. of secret encryption keys	<= nodes	3 * nodes	= nodes	= nodes
No. of user decryption keys	<= his nodes	> His nodes	One secret key	One private key
Secure Channel used	With each document	With each document	With each document	With each update
User's Encrypted document size	Whole document	Whole document	Whole document	Whole document
System update	Expensive	Expensive	Expensive	Limited cheap
Temporal access	Yes	No	No	No
Access delegation	No	No	No	No

2. MATERIALS AND METHODS

2.1. Key Management Scheme

Any push-based XML access control solution must ensure the confidentiality and integrity of distributed data (XML documents, policies and secret keys). In this study, the Identity Based Encryption (IBE) is used which is a public key cryptography system but the public key is an arbitrary string. Hence, using the IBE provides us with several advantages such as: (i) supporting the temporal access by generating temporal public/private keys for a user from his identity plus access period (for example "Sam+2009"); (ii) supporting delegable access by generating delegated public/private keys from the delegated user identity plus access period plus delegation period (e.g., "Sam+2009+May"); and (iii) simplifying the key management task because in IBE the public key of user is his identity and there is no need to use a PKI to provide each user with a public key certificate. In this research, the public/private keys of users are generated based on the IBE scheme (Boneh and Franklin, 2001) which is the first efficient, secure and widely used IBE schema (Gagne, 2003). The public/private keys are generated as follows: (i) generating an ID, which is a byte array generated from the string value of the user's identity; (ii) generating the public key (Q_{ID}) by mapping the ID to a point on an elliptic curve. In fact, the IBE is based on elliptic curve cryptography; and (iii) generating the private key (sQ_{ID}) by computing $sQ_{ID} = \times Q_{ID}$.

An Advanced Encryption Standard (AES) is used for encrypting a document's nodes or portions with 128-bit secret keys and then the encryption secret keys are distributed by encrypting them using an IBE-based key distribution scheme proposed by Du *et al*. (2005). A General Public Key (GPK) is calculated (as a sum) from the public keys of the authorized users and used for encrypting the secret encryption keys, which are then decrypted only by the right users. The Du *et al*. (2005) scheme is secure unless there are more than two users (Chien, 2007). For this reason, we initialized the GPK with two random public keys before adding to it the public keys of the authorized users.

2.2. Policy Enforcement Mechanism (PEM)

The Policy Enforcement Mechanism (PEM) encrypts the XML documents based on the specified policies and then publish the encrypted document to right users based on the received XML documents and policy certificates. For more information about the policy certificates (user, role and permission certificates) we refer to our proposed policy language in (Halboob *et al*., 2008). The PEM handles two components a Dynamic Key Management Table (DKMT) and XML Document Encryption/Decryption Process (XDEDP).

2.3. Dynamic Key Management Table (DKMT)

The DKMT contains several fields extracted from the policy certificates. As shown in **Table 2**, these fields are *Ucert_SN* (user certificate unique serial number), *User_address* (to which encrypted document will be published e.g., Email), *U_name* (user name), *User_Pkey* (user public key), *Role_name* (Role name assigned to user), *Role_GPK* (role general public key calculated from the public keys of users assigned to this role). The DKMT helps on reducing the system update cost since any required update can be achieved by updating the DKMT only without re-marking XML documents and/or re-distribute the new secret keys to users. For example, adding/removing a user requires only adding/subtracting his public key from the GPKs of his related roles. Also, this table helps on improving the system performance during publishing the document to users as there is no need for parsing and processing the published XML document for marking each node with its related policies and secret keys. Where processing such XML documents is more expensive than the table in term of memory usage, since XML document are represented at run time using a Document Object Model (DOM) and the DOM size in the system memory is larger (about 5 to 10 times) than the XML document size (Wang and Tan, 2001).

Table 2. Dynamic Key Management Table (DKMT)

Ucert_SN	User_address	U_name	Acc_Period	User_Pkey	Role_name	Role_GPK
U12	User1@	User1	2008	Q_{user1}	Full access	Q_{r1}
U234	User2@	User 2	May_2008	Q_{user2}	Database Issues	Q_{r2}
…..	…..	…..	…..	…..	…	…

```
INPUT: XML Document; Recived users list UL and their roles RL from DKMT
OUTPUT: Encrypted XML Document ED
   Let Enc_Document=null; Let Roles_List= Roles of users in UL// from the DKMT
   For each Role in Roles_List
     Let i=1;
     Let Permissions_list=null;
     Let Nodes_List=null
     Permissions_list=Permssions of this Role// from the role certificate
     While i<=n
         Nodes_List= Nodes_List+ nodes of Permissions_list[i]
     EndWhile
     Filt_Doc = YFilter (XML Document, Nodes-list)
     s= Generate 128-bit secret key //generating a new secret s
     En_Role_Nodes = Enc (Filt_Doc) with s //Encrypting the role's nodes with s
     s=Enc (s) with a GPK of this role
     Enc_Document = Combine (Enc_Documet + En_Role_Nodes)
   EndFor
 Publish Enc_Documet to all users in the UL
```

Fig. 1. An algorithm for XML document encryption

```
INPUT: Enc_Document, User Certificate, User Private Key
OUTPUT: Dec_Document
    Let Roles_List= Roles of users in User certificate
    Let Dec_Document=null
    For each Role in Roles_List
        Let i=1; Let Permissions_list=null;
        Let Nodes_List=null
        Permissions_list=Permssions of this Role// from the User certificate
        While i<=n
             Nodes_List= Nodes_List+ nodes of Permissions_list[i
        EndWhile
        Let Permissions_list=null
         Filt_Doc = YFilter (Enc_Document, Nodes-list) //filtering the nodes
              related to this Role
        s=Dec (s) with a User Private Key
        Dec_Role_Nodes = Dec (Filt_Doc) with s
        Dec_Document = Combine (Dec_Documet + Dec_Role_Nodes)
    EndFor
Rerturn Dec_Documet
```

Fig. 2. An algorithm for XML document decryption

Since the relationship between the users and their roles is a many-to-many relation, so the DKMT is separated into three sub-tables namely Key Table (KT), Role Assignment Table (RAT) and Role Key Table (RKT). Due to our limited space provided here we leave the describing of these sub-tables and their update cases.

2.4. XML Document Encryption/Decryption Process (XDEDP)

The XML Document Publishing Process (XDPP) is used for encrypting and publishing the XML document to users based on the data stored in the DKMT. The algorithm of encrypting the XML document is shown in **Fig. 1**.

As shown by the encryption algorithm in **Fig. 1**, the number of the required secret encryption keys is reduced to the number of roles, instead to the number of nodes as used in the related works. The role is a set of permissions and each permission is a set of nodes. Also, our encryption algorithm allows the publisher to publish the data of only selected set of users, instead of publishing the data of all users and the YFilter technique (Diao *et al*., 2002) is used for filtering the nodes of the selected users.

Figure 2 shows the document decryption algorithm used by the user which needs only to handle his permanent private key to decrypt all received documents. He is also able to reduce the size of the decrypted document by removing the unrelated nodes using YFilter technique (Diao *et al*., 2002).

3. RESULTS AND DISCUSSION

To the best of our knowledge, our solution is the first push-based XML access control policy enforcement that provides a user with delegable access and a reduced decrypted document size. In addition, our model has improved results in several criteria, as shown in **Table 2** and is ordered based on their properties as follows: (1) the temporal access only requires giving the user a temporal IBE public/private key pair.

The provided temporal access by Author-X in (Bertino *et al*., 2001; 2003; 2007) requires giving the users several temporal secret keys (equal to his policies] and specifying temporal policies in such a way that a policy is re-specified several times if it is assigned to different users with different access periods. This weakness increases the number of policies handled and, as a result, affects both the access control system and management scalability; (2) the user's decrypted document is reduced by enabling the user to filter his unrelated nodes, this helps to save the user's storage capacity, which is normally less that the server's storage; (3) the system update cost is reduced with a dynamic (changeable) number of users, unlike model (Zhang *et al*., 2005; 2004), which reduces the system update cost but only with a group (limited number) of users; (4) the use of the secure channel is reduced to only once for each user during his subscription lifetime. In contrast, Zhang's model (Zhang *et al*., 2005; 2004), works only with a group of users and increasing the group size makes it necessary to give each user a new public/private key and to use the secure channel again for distributing the new public/private keys; (5) as achieved by (Zhang *et al*., 2005; 2004), our model gives each user only a permanent private decryption key. However, model (Zhang *et al*., 2005; 2004), affects the user by making it necessary to use a new private key if the size of the user's group is increased and (6) the number of required secret encryption keys is widely reduced to the number of roles and the role is a set of permissions, which is also a set of nodes. In Author-X (Bertino *et al*., 2001; 2003; 2007), reducing the number of required secret encryption keys involves the system in a searching process for matching and marking the nodes that have the same access with the same key. This process affects the system efficiency and increases the encrypted document size by adding metadata to each node during the marking step.

4. CONCLUSION

In this study, we proposed a policy enforcement mechanism for a push-based access control model. The proposed mechanism provides users with temporal and delegable access and reduces the system update cost size of a user's decrypted document the use frequency of the secure channel is also reduced to only once for each user during his subscription period. The secret encryption keys required for encrypting the nodes of the published document are widely reduced, while the user needs only his private decryption key to decrypt all of his related nodes. In this research the published document is encrypted into a number of documents equal to the number of roles. As future work, there is a need to apply an approach that enables a user to combine his roles' decrypted documents into one document that has the same structure as the original document. We have implement and evaluate the performance of this mechanism which cannot be listed here because of limited space.

5. REFERENCES

Bertino, E., B. Carminati, E. Ferrari and G. Mella, 2003. Author-X-A system for secure dissemination and update of XML documents. Data. Netw. Inform. Syst., 2822: 66-85. DOI: 10.1007/978-3-540-39845-5_7

Bertino, E., E. Ferrari, F. Pad and L. Provenza, 2007. A system for securing push-based distribution of XML documents. Int. J. Inform. Sec., 6: 255-284. DOI: 10.1007/s10207-007-0020-3

Bertino, E., S. Castano and E. Ferrari, 2001. Author-x: A Comprehensive system for securing XML documents. IEEE Internet Comput., 5: 21-31.

Bertino, E., S. Castano, E. Ferrari and M. Mesiti, 1999. Controlled Access and Dissemination of XML Documents. Proceedings of the 2nd International Workshop on Web Information and Data Management, Nov. 02-06, ACM New York, USA., pp: 22-27. DOI: 10.1145/319759.319770

Boneh, D. and M. Franklin, 2001. Identity-based encryption from the weil pairing. Proceedings of the 21st Annual International Cryptology Conference on Advances in Cryptology, Springer Berlin Heidelberg, Santa Barbara, California, USA., pp: 213-229. DOI 10.1007/3-540-44647-8_13

Bouganim, L., F.D. Ngoc and P. Pucheral, 2008. Dynamic access-control policies on encrypted data. ACM Trans. Infor. Syst. Sec. DOI: 10.1145/1284680.1284684

Chen, Y., 2007. An Efficient Algorithm for Tree Mapping in XML Databases. J. Comput. Sci., 3: 487-493.

Chien, H.Y., 2007. Comments on an efficient ID-based broadcast encryption scheme. IEEE Trans. Broadcast., 53: 809-810. DOI: 10.1109/TBC.2007.908331

Crampton, J., 2004. Applying hierarchical and role-based access control to XML documents. Proceedings of the Workshop on Secure web Service, (WSS' 04), ACM New York, USA., pp: 37-46. DOI: 10.1145/1111348.1111353

Crampton, J., 2006. Applying hierarchical and role-based access control to XML documents. Int. J. Comput. Sci. Syst. Eng., 21: 325-338.

Diao, Y., P. Fischer, M.J. Franklin and R. To, 2002. YFilter: Efficient and Scalable Filtering of XML Documents. Proceedings of the 18th International Conference on Data Engineering, Feb. 26-Mar. 01, IEEE Xplore Press, San Jose, CA., pp: 341-342. DOI: 10.1109/ICDE.2002.994748

Du, X., Y. Wang, J. Ge and Y. Wang, 2005. An ID-based broadcast encryption scheme for key distribution. IEEE Trans. Broadcast., 51: 264-266. DOI: 10.1109/TBC.2005.847600

Gagne, M., 2003. Identity-based encryption: A survey. RSA Laboratories Cryptobytes, 6: 10-19.

Halboob, W., A. Mamat and R. Mahmod, 2008. A distributed push-based XML access control model for better scalability. Proceeidngs of the 1st International Conference on Distributed Framework and Applications, Oct. 21-22, IEEE Xplore Press, Penang, pp: 20-26. DOI: 10.1109/ICDFMA.2008.4784409

Ko, H.K., M.J. Kim and S. Lee, 2007. On the efficiency of secure XML broadcasting. Inform. Sci., 177: 5505-5521. DOI: 10.1016/j.ins.2007.05.020

Miklau, G. and D. Suciu, 2003. Controlling access to published data using cryptography. Proceedings of the 29th International Conference on Very Large Data, ACM., VLDB Endowment, pp: 898-909.

Mu, Y. and V. Varadharajan, 2001. Robust and Secure Broadcasting. Proceedings of the Second International Conference on Cryptology in India: Progress in Cryptology, Dec. 16-20, Springer Berlin Heidelberg, India, pp: 223-231. DOI: 10.1007/3-540-45311-3_21

Wang, Y. and K.L. Tan, 2001. A Scalable XML Access Control System. Proceedings of the 10th International World Wide Web Conference, May 1-5, Hong Kong, China.

Zhang, J., V. Varadharajan and Y. Mu, 2004. Securing XML document sources and their distribution. Proceedings of the 18th International Conference on Advanced Information Networking and Applications, Mar. 29-31, IEEE Xplore Press, pp: 562-56. DOI: 10.1109/AINA.2004.1283969

Zhang, J., V. Varadharajan and Y. Mu, 2005. Secure Distribution and access of XML documents. Int. J. High Perform. Comput. Network., 3: 356-365. DOI: 10.1504/IJHPCN.2005.009423