RBAC and XML Security in Adhoc Networks

Qurban A. Memon*, Shakeel Khoja**
*Associate Professor, EE Department, UAE University, Al-Ain 17555, UAE
**Associate Professor, Department of Computer Science and Engineering, Bahria University, Pakistan

Abstract - **As adhoc networks are becoming increasingly important for variety of applications, so are the rules and specifications for their formations and operations. In this paper, we describe how roles of participating devices can be framed to facilitate access and formation of an adhoc network. Further, we propose XML framework for secured file transfer in a typical case where adhoc network nodes exchange files and data among themselves. We also show corresponding XML pseudo-code in the light of world-wide-web consortium (W3C) guidelines.**

I. INTRODUCTION

A mobile adhoc network (MANET) [1-3] provides a communication environment that is characterized by dynamic changes in the topology and in the availability of resources. More research work has, recently, been reported in literature, which addresses issues of security across different aspects of adhoc networks. For example, the authors in [4-6] address the weaknesses of current routing protocols in adhoc networks with respect to security and highlight the robustness of the mechanisms proposed in their work. In [7], the authors have addressed security issues in an environment where devices remain in contact with each other for longer duration. For this purpose, a fully distributed approach has been proposed to secure long term communities of devices. Buchegger and Le Boudec [8] have proposed a technique called CONFIDANT (Cooperation of Nodes, Fairness In Demand Adhoc NeTworks) that primarily aims at detecting malicious nodes by means of combined monitoring and reporting and establishing routes by avoiding misbehaving nodes. The weakness lies in detection process where node identity is assumed to be persistent. This may lead to spoofing attacks. A related work can also be found in [9], where authors have addressed a subset of all threats in adhoc networks and have proposed short of a comprehensive answer to the security problem.

Typically, collaborations among the participants that form an adhoc network cannot be set up because they do not trust each other to use their respective services and resources [10-11]. Therefore, there is a need for explicit specification of policies for each activity. A large number of enterprises have recently started to explore Internet based workflow management systems to help improve their services and decision-making processes [12]. There has been a number of access control models discussed in literature for various objectives [13-14]. Among them, the RBAC model is gaining attention as a generalized approach and provides several advantages. Under the RBAC framework, users are granted membership into roles based on their responsibilities in the organization. User membership into roles can be revoked easily and new memberships established as job assignments dictate. Role-Based Access Control (RBAC) models [15-16] are receiving increasing attention as a generalized approach to access control. Thus, a role is a collection of permissions (or operations on a set of objects) determined by the system, based on the users organizational activities and responsibilities. Furthermore, RBAC has shown to be policy neutral [17] and supports security policy objectives, and static and dynamic separation of duty constraints [16]. Moreover, RBAC offers flexibility with respect to different security policies and in fact [18] shows that RBAC can be configured to enforce mandatory and discretionary access control policies.

The eXtensible Markup Language (XML) is regarded generally as having promise of becoming established as the general purpose framework for enabling transfer of data amongst heterogeneous environments. It is emerging as a useful platform-independent data representation language and is growing in proportion to the spreading speed of e-commerce, which requires a policy for providing a safer security service for exchanging e-documents within e-commerce. It is of interest therefore to analyze how suitable it may be once details of applications requirements and constraints are taken into account. Further, it becomes necessary to consider the issue of XML access control and document security in different network environments, for example adhoc network environment.

A lot of research work has been carried out in the area of XML, its specification development, and security features etc. For example in the area of XML, the authors in [19] propose a construct that locates related nodes in an instance of an XML data model, independent of a specific structure. High performance XML parsers can be prepared using parser generation and compilation techniques [20]. In that, parsing is integrated with Schema-based validation and deserialization, and the resulting validating parsers are shown to be as fast as or in many cases significantly faster than traditional non-validating parsers [20].

In the area of security, XKMS (XML key management specification) is one of the XML's security specifications that define the protocol for distributing and registering public keys for verifying digital signatures and enciphering e-documents of e-commerce applications with various and complicated functions. The authors in [21] discuss XKMS-based key management system architecture and a service model using Java crypto technologies and XML security mechanisms. XML digital signature capability has also been discussed in [22], where authors describe a solution to add XML digital signature using a signature server implemented as a web service without modifying XML based systems. In [23], the authors explore eXtensible Stylesheet Language (XSL), whose document transformation component (XSLT) may well have sufficient functionality to perform all reasonable cryptographic transformations to deliver a desired level of document security. The authors in [24] describe developments in

semantic Web and then provide an overview of secure semantic Web. In particular XML security, RDF security, and secure information integration and trust on the semantic Web are addressed. Another work on Web and e-commerce applications security has been reported in [25], where authors discuss, in particular, access control policies, workflow security, XML security and federated database security issues pertaining to the Web and e-commerce applications. In emerging applications, a similar work is found in [26], where future directions for data and applications security that include secure semantic Web, XML security and applications such as bioinformatics, peer-to-peer computing, and stream information management are addressed. In [27], the authors discuss concept of cryptographically secured, XML based Security Labels using a guard prototype for file transfer and web services based applications. Another approach related to XML access control is reported in [28], where storage of the accessibility information is based on the compressed accessibility map (CAM), and further improved by integrating multiple CAMs into an integrated CAM (ICAM).

II. PROPOSED APPROACH

The formation of an adhoc network requires that authentication and server devices be present to initiate formation of an adhoc network. The joining of a device is to be authenticated by a server owned by the organization. Such network architecture is depicted in Figure 1. It consists of a group of devices forming an adhoc network through an authentication server. We propose a multi-channel model for accessing services and information interchange among users. The objective is to define and categorize types of data transfer pertaining to general and role specific use respectively for sake of easiness and simplicity. The universal description, discovery, and integration (UDDI) channel is proposed to include registry information about the groups, given by the central server and propagated by the coordinator device. Each entry in the UDDI channel is identified by Key_i, and information within the channel is customized to fit wireless environments. The session channel contains the description of each session, and information within the session channel is indexed with a service key to enable better access performance. The data channel is used to transfer data among network devices. Whenever a device enters an adhoc network, it downloads UDDI channels content to its device and store it for later use. Caching it avoids frequent access to the channel, and minimizes power consumption of the devices as well. Thus, the organizations need to empower mobile user devices with the ability to:

- Discover session and data channel(s)
- Find out the way to invoke the session (like which input parameters are required).

The impact of using a data buffer, as shown in Figure 1, is to improve cache retrieval. The objective of coordinator buffer is to share and cache those data frequently used by all mobile users. This reduces the amount of disk accesses in central databases. At the end of transaction execution, the device has to perform buffer write to store updated data. This in turn invalidates obsolete data in the buffer. The authentication server may be empowered to supervise many adhoc network(s). In case, if authentication device leaves and another takes over, a new broadcast from authentication server is to be initiated to let all devices know that there is a new authentication server. This however, will require all devices to re-authenticate.

Roles essentially partition database information into access contexts. Methods associated with a database object, also partition the object interface to provide windowed access to object information. By specifying that all database information is held in database objects and authorizing methods to roles, we achieve object interface distribution across roles. By authorizing different users to the different roles, we can enforce both the order of execution on the objects and separation of duty constraints on method execution. Because of space limitations in mobile devices (like laptops etc.), data is proposed to be at database server, where single or distributed databases can be placed, and then caching of events, registries and other services can be allowed on individual devices.

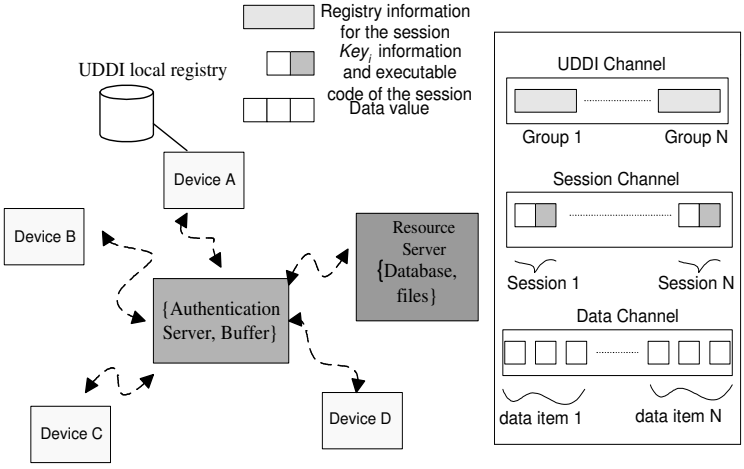

Figure 1: Adhoc Network Infrastructure with multi channels

A. Access Security

The policy based design of adhoc groups is the idea that groups are defined around objects and that objects can be hierarchically combined. Objects might include people, teams, locations, tasks, projects, or meetings, data (such as files or database tables), and resources (such as printers, scanners, or displays). All objects are uniquely identified by resource identifiers akin to uniform resource indicators (URIs) and will need to be maintained on a server and on individual devices as needed to provide redundancy when connectivity is unavailable. Based on these guidelines, the architecture for access control in adhoc networks is proposed of four components as shown in Figure 2. The components are: profile management and membership management (combined as user management), protocol management, policy enforcement and an event service. The framework runs on every user's device. The profile management component maintains the user's credentials, such as key certificates and stores, and attributes certificates. Users can manage their credentials and device

settings through user management interface. In addition, this component also maintains the user's preferences on which communities the device should automatically join. The membership management component exposes the user management interface to the application level, so that applications can initiate the establishment of a new community, search for communities, as well as joining particular communities. Through this interface, the user can register the services that it is providing to other participants. The membership management component is also responsible for checking the authenticity of the doctrines and enforcing them by extracting and distributing the policy instances to various enforcement components. Lastly, the event service collects and aggregates events and subsequently forwards them to the policy enforcement, e.g. the triggering of the execution of obligation policies. System events are forwarded to the protocol management, so that appropriate protocols can be performed. Events regarding the discovery of new communities are forwarded to the membership management component.

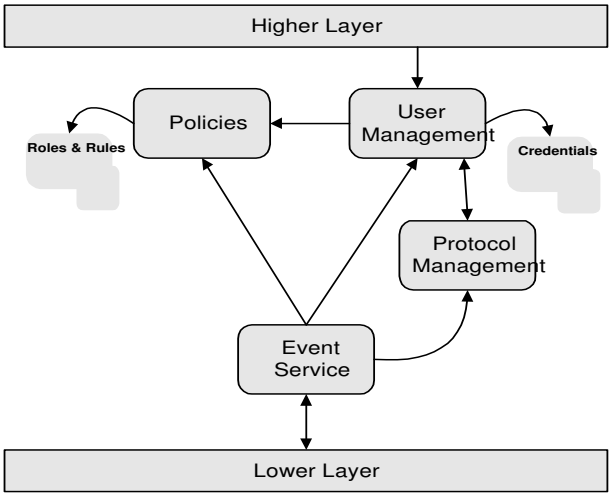

Figure 2: The Acess Control Framework for Roles

B. Access algorithm for devices

Below, we present an algorithm with the main steps that need to be performed to execute the network access service. Adhoc users generally start by looking for service on their category and role. They may look for services initiated by their parent organization and are specific to their roles. The tuning and access to a channel is to be performed by an appropriate access method.

Algorithm execute-service:

/*executed whenever device sniffs an adhoc network*/

Begin:

a. Find adhoc service having a given category in the local UDDI directory
b. Select a service and retrieve its service key Key_i and compare with the key stored in the event service.
c. Retrieve the frequency of the service channel
d. Listen to the service channel
e. Download the description of the service having Key_i as the service key.
f. Based on the service key Key_i, determine input parameters (from user management) to initiate access to the network.
g. Proceed to login to the network.
h. After successful login, retrieve the frequency of the data channel
i. Download role specification for the device and store it in event service
j. Execute the service or exchange data with other users on the data channel of the network.

End

C. XML Framework

There are various traditional techniques that use wrapping over an XML document to provide document security; however none of them can be embedded within the document. An essential requirement of proposed XML security framework is that it should work naturally with content created using XML. The objective is to guarantee the aspects of integrity, confidentiality, and accountability (key management). This is achieved by fitting together the ideas of the XML encryption and XML digital signature in the light of the specifications provided by world wide web consortium (W3C) [29]. We define the framework as formation of components. These components are independent and provide atomic operations that are needed while implementing a secure XML based application. The components should be designed with respect to the level they are categorized into, such as level 1 and level 2 components.

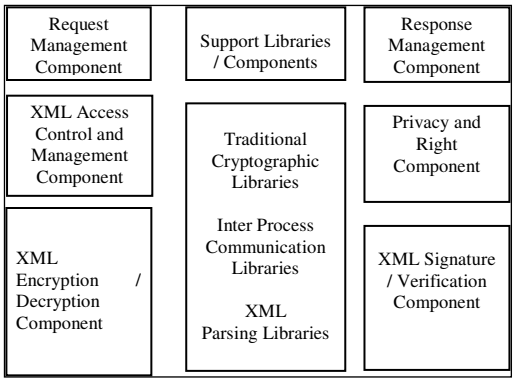

Figure 3: XML Encryption and Signing Components

At the first layer are level 1 components consisting of all the basic level functionality that is required by any security system. The level 1 components will give an interface to the next level components to use the traditionally available methods for security and also provide XML parsing capabilities to the higher level components, such as parsers, request/response management components and inter-communication components. The level 2 components are XML transformation components, key management components, encryption/decryption components, signature/validation components and access control components, forming the core of the framework. These components provide the methods and their implementations that are defined or outlined in the various XML security related specifications provided by the W3C.

XML Encryption and Signing Component: In the encryption and signing process, there can be a possible need for counter signature, or partial encryption. The framework proposes that it is better to perform signature or encryption by processing input XML along with a stencil/template that specifies a signature or encryption skeleton, the way to use transformation component, the usage methods for traditional cryptographic/algorithm components, and the way to interface with XML key management component for key selection process. This stencil document will be an XML document itself with similar structure as the desired result but some of the nodes will be left empty and will be filled by XML encryption/signature components after performing relevant computations. XML Security Framework gets the key for signature/encryption from the key manager in the key management component using the information from the stencil document, does necessary computations and puts the results in empty nodes of the given stencil, as shown in Figure 3. Signature or encryption component controls the whole process and stores the required temporary data. Since the Stencil information is also a XML file, it might be created in advance and saved in a file and can be given to the application as an input otherwise the security framework Application Program Interface (API) will have to generate a stencil by gathering information by itself. This logic allows application to create stencils without using XML Security Framework functions. Also in some cases stencil should be inserted in the signed or encrypted data (for example, if you want to create an enveloped or enveloping signature). Signature verification and data decryption do not require template because all the necessary information is provided in the signed or encrypted document.

XML document model: Proposed XML document model, enabling role-wise security in an XML document is shown in Figure 4. XML Digital signatures are used to lock/encrypt a selected part of an XML document. They also provide end-to-end message integrity guarantees, and can also provide authentication information about the originator of a message. An XML signature would define a series of XML elements that could be embedded in, or otherwise affiliated with, any XML document. It would allow the receiver to verify that the message has not been modified from what the sender intended. Format of an XML signature is shown in Figure 5.

Components of Digital Signature: The Id attribute, shown in Figure 5 allows a document to contain multiple signatures, and provides a way to identify particular instances. Multiple signatures are used for accessing the data with different levels of rights. The SignedValue element contains the actual signature, which is a base64-encoded data, as defined by XML DSIG specifications [30]. The ds:object (Figure 5) element is used to hold metadata or additional information that is considered as a 'property' of the digital signature. For example, a timestamp for when the signature is generated will be considered as a property of the digital signature.

The contents of ds:SignedInfo can be divided into two parts, information about the Signature value, and information about the application content, as shown in the code given in Figure 6, defining attributes regarding canonicalization (C14N) method, signature method and the reference elements used. Reference element contains the digest of the content, an indication that, how digest was generated and a specification of how the content should be transformed before the digest is generated.

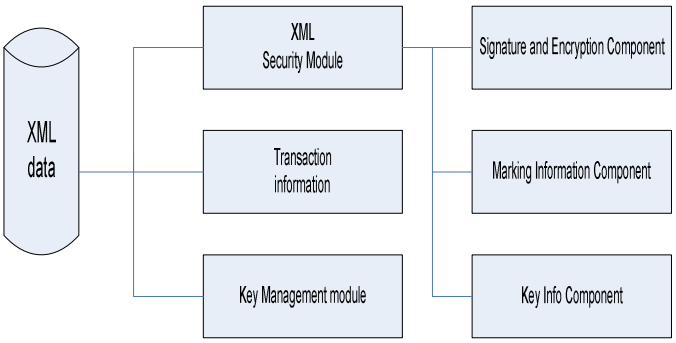

Figure 4: Key components of XML security enabled document.

```
<element name="Signature" type="ds:SignatureType"/>
  <complexType name="SignatureType">
    <sequence>
      <element ref="ds:SignedInfo"/>
      <element ref="ds:SignatureValue"/>
      <element ref="ds:KeyInfo" minOccurs="0"/>
      <element ref="ds:Object" minOccurs="0" maxOccurs="unbounded"/>
    </sequence>
    <attribute name="Id" type="ID" use="optional"/>
  </complexType>
```

Figure 5 : Format of a Digital Signature

```
<element name="SignedInfo" type="ds:SignedInfoType"/>
  <complexType name="SignedInfoType">
    <sequence>
      <element ref="ds:CanonicalizationMethod"/>
      <element ref="ds:SignatureMethod"/>
      <element ref="ds:Reference" maxOccurs="unbounded"/>
    </sequence>
    <attribute name="Id" type="ID" use="optional"/>
  </complexType>>
```

Figure 6: SignedInfo component

Marking Component: Marking components are used to mark components, requirements, and configurations of the required part of the document. The tags used for marking components are role, role-hint, version, lifecycle-handler and instation-strategy of the component to be used.

Key Info Component: Key Info component is used to identify the signer, or at least the key that generated the signature. Key Info component also stores the information of the key that is used to protect the digest from being modified. Figure 7 shows a pseudo XML code for Key Info element, providing a wide variety of key types and key infrastructures.

III. CONCLUSIONS

We have described role based access and XML based data transfer within an adhoc network. This model helps in providing security at two levels: authentication based on role, and XML embedded data transfer. As XML

model specification is evolving together with commercial products for embedding roles within databases, both of these independent levels can be merged within one server or an application. This way, security within an adhoc network can further be simplified. Currently, we are exploring on these lines.

IV. REFERENCES

[1]. Y. Hu et al. Ariadne "A Secure On-demand Routing Protocol for Ad Hoc Networks". *Proceedings of the* 8th *ACM International Conference on Mobile Computing and Networking*, September 2002.

[2]. F. Stajano. "The Resurrecting Duckling –What Next?" *Proceedings of the 8th International Workshop on Security Protocols*, 2000.

[3]. F. Stajano and R. Anderson "The Resurrecting Duckling: Security Issues for Ad-hoc Wireless Networks", *Proceedings of the 7th International Workshop on Security Protocols*, 1999.

[4]. P. Papadimitratos, Z. Haas, "Secure Routing for Mobile Adhoc Networks", in *Proceedings of the SCS Communication Networks and Distributed Systems Modeling and Simulation Conference*, *CNDS* 2002.

[5]. S. Bhargava and D. P. Agrawal. "*Security enhancements in AODV protocol for wireless adhoc networks*", Vehicular Technology Conference, 2001.

[6]. S. Bhargava, D. Agrawal, "Scalable Security Schemes for Ad Hoc Networks", *IEEE Milcom 2002*, Anaheim, California, October 7-10, 2002.

[7]. N. Prigent, J.-P. Andreaux, C. Bidan, O. Heen, "Secure Long Term Communities in AdHoc Networks", 1st *ACM workshop on Security in Ad hoc and Sensor Networks* (SASN), Fairfax, North Virginia, U.S.A., 2003.

[8]. S. Buchegger and J-Y Le Boudec, "Performance Analysis of the CONFIDANT Protocol: Cooperation Of Nodes – Fairness In Distributed Adhoc NeTworks", In *Proceedings of IEEE/ACM Workshop on Mobile AdHoc Networking and Computing (MobiHOC)*, Lausanne, CH, June 2002.

[9]. Refik Molva, Pietro Michiardi, "Security in Ad Hoc Networks", *Personal Wireless Communications*, *IFIP-TC6 8th International Conference*, pp. 756-775, Italy, 2003.

[10]. Y. Zhang and W. Lee "An Integrated Environment for Testing Mobile Ad-Hoc Networks". In *3rd ACM Symp. on Mobile Ad Hoc Networking and Computing (MobiHoc)*, June 2002.

[11]. L. Zhou and Z. J. Haas. "Securing Ad-Hoc Networks". *IEEE Network Magazine*, Vol. 13, No. 6, November/December 1999.

[12]. Dan C. Marinescu, *Internet-Based Workflow Management: Toward a Semantic Web*, ISBN: 0-471-43962-2, Wiley Publishers, April 2002.

[13]. J. Doshi, W. Aref, A. Ghafoor, and E. Spafford, "Security Models for Web-Based Applications", *Communications of the ACM*, Vol. 44, No. 2, pp. 38-44, February 2001.

[14]. R. Sandhu, "Lattice based access control models", *IEEE Computer*, 26, 11, 1993.

[15]. Sandhu, R.S., Coyne, E.J., Feinstein, H.L., Youman, C.E. "Role-based access control models". *IEEE Computer,* 29 (1996) pp. 38-47.

[16]. Ferraiolo, D.F., Sandhu, R., Gavrila, S., Kuhn, D.R., Chandramouli, R. "Proposed NIST standard for role-based access control". *ACM Transactions on Information and System Security* (TISSEC) 4 (2001) pp. 224-274.

[17]. Bertino, E., Bonatti, P.A., Ferrari, E. "TRBAC: A temporal role-based access control model". *ACM Transactions on Information and System Security* 4 (2001) pp. 191-223.

[18]. Osborn, S., Sandhu, R., Munawer, Q. "Configuring role-based access control to enforce mandatory and discretionary access control policies". *ACM Transactions on Information and System Security* (TISSEC) 3 (2000), pp. 85-106.

[19]. S. Zhang, C. Dyreson, Symmetrically exploiting XML, Proceedings of the 15th International Conference on Word Wide Web (WWW '06), May 23-26, 2006, pp. 103-111.

[20]. Kostoulas, M. G., Matsa, M., Mendelsohn, N., Perkins, E., Heifets, A., and Mercaldi, M., XML Screamer: An Integrated Approach to High Performance XML Parsing, Validation and Deserialization, Proceedings of the 15th International Conference on World Wide Web (WWW '06). pp. 93-102.

[21]. N. Park, K. Moon, and Sungwon Sohn, XML Key Management System for Web-Based Business Applications, IEEE/IFIP Network Operations and Management Symposium (NMOS 2004), 19-23 April, 2004, Vol. 1, pp. 903-904.

[22]. Takase, T. Uramoto, N. Baba, K., XML Digital Signature System Independent of Existing Applications, Proceedings of Symposium on Applications and the Internet (SAINT) Workshops, Jan. 28- Feb. 01, 2002, pp. 150-157.

[23]. Bartlett, R.G. Cook, M.W., XML Security using XSLT, Proceedings of the 36th Annual Hawaii International Conference on Systems Sciences, 6-9 Jan 2003, 6 pp. on CDROM.

[24]. Thuraisingham, B., Security Issues for the Semantic Web, Proceedings of 27th Annual International Computer Software and Applications Conference (COMPSAC 2003), 3-6 Nov. 2003, pp. 633 – 638.

[25]. Thuraisingham, B. Clifton, C. Gupta, A. Bertino, E. Ferrari, E., Directions for Web and e-commerce Applications Security, Proceedings of Tenth IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET ICE 2001), 20-22 June 2001, Cambridge, MA, pp. 200 – 204.

[26]. Thuraisingham, B., Data and Applications Security: Developments and Directions, Proceedings of 26th Annual International Computer Software and Applications Conference, (COMPSAC 2002), 26-29 Aug. 2002, pp. 963 – 965.

[27]. Thummel, A. Eckstein, K, Design and Implementation of a File Transfer and Web Services Guard Employing Cryptographically Secured XML Security Labels, IEEE Information Assurance Workshop, June 21-23, 2006, pp. 26 – 33.

[28]. Mingfei, J., Ada, F., Integration and Efficient Lookup of Compressed XML Accessibility Maps, *IEEE Transactions on Knowledge and Data Engineering*, Vol. 17, Issue 7, July 2005, pp. 939 – 953.

[29]. http://lists.w3.org/Archives/Public/xml-encryption/

[30]. IETF/W3C XML-DSig Working Group, http://www.w3.org/TR/xmldsig-core/:


```
<element name="KeyInfo" type="ds:KeyInfoType"/>
 <complexType name="KeyInfoType" mixed="true">
  <choice maxOccurs="unbounded">
   <element ref="ds:KeyName"/>
   <element ref="ds:KeyValue"/>
   <element ref="ds:RetrievalMethod"/>
   <element ref="ds:X509Data"/>
   <element ref="ds:PGPData"/>
   <element ref="ds:SPKIData"/>
   <element ref="ds:MgmtData"/>
  </choice>
 <attribute name="Id" type="ID" use="optional"/>
 </complexType>
```

Figure 7: Key Info Component