Role-based access to facilities lifecycle information on RFID tags

Ali Motamedi, Rakesh Saini, Amin Hammad & Bo Zhu
Concordia University, Montreal, Quebec, Canada

Abstract

Radio Frequency Identification (RFID) based systems have been used in different applications in the Architecture, Engineering, Construction, Owner and Operator (AECOO) industry. Applications are mainly designed for specific lifecycle stage and serve the needs of only one of the stakeholders. This would increase the cost and the labor for adding and removing tags and eliminate the chance of using shared resources. In this paper, the usage of tags permanently attached to components is proposed where tags are attached at an early stage of the lifecycle and the memory of the tags is used during the lifecycle by different stakeholders. Using a Building Information Model (BIM) database is proposed for tackling the interoperability issues allowing different users to access and share the data. It is also proposed to securely and efficiently store data on RFID tags in ifcXML format using multi-level encryption together with role-based access control. To explore the technical feasibility of the proposed approach, a case study has been implemented and tested at Concordia University.

Keywords: RFID, lifecycle management, BIM, RFID security.

1 Introduction

Efficient Information sharing and exchange between various players in the Architects, Engineers, Constructors, Owners and Operators (AECOO) industry is an evident need. The Building Information Model (BIM) is emerging as a method of creating, sharing, exchanging and managing facilities components information throughout the lifecycle between all the stakeholders (Isikdag et al., 2007). Radio Frequency Identification (RFID), on the other hand, has emerged as an automatic data collection and information storage technology and has been used in different applications in the AECOO industry (Jaselskis and El-Misalami, 2003). RFID tags are mainly used to facilitate operations at specific lifecycle stages to store information related to specific operations and to serve the needs of only one of the stakeholders. In many applications, the tags are removed at the end of the operation and reused. This would increase the cost and labor for adding and removing different tags and eliminate the chance of using shared resources among the stakeholders (Motamedi and Hammad, 2009).

The lifecycle of a building can be divided into different stages where each stage is generally managed independently while exchanging partial information with other stages. The information related to each component should be tracked separately throughout the lifecycle. Furthermore, the information should be in a convenient format and stored at a suitable location to enable all the stakeholders to access throughout the lifecycle. Centrally stored information that is accessible over a
computer network is a solution for data access. However, having real-time access to information could be difficult since reliable connections to the central data storage may not be always available.

The objectives of this paper are: (1) to investigate the idea of components lifecycle management using RFID tags populated with BIM data; (2) to investigate the security and privacy solution for data stored on RFID tags; and (3) to demonstrate the feasibility of the proposed approach through a real-world case study.

2 Proposed Approach

The usage of permanently attached RFID tags to components is proposed where tags are attached to components at an early stage of the lifecycle and the memory of the tags is used for various applications. Having information storage on the components allows different stakeholders to use the memory for data storage and information handover to other users. In the proposed approach, the information on the tags is stored in ifcXML standard format representing chunks of the BIM as a distributed database. The memory on the tags is divided into different logical sections, which will be gradually filled with updated information based on the current status of the component.

In addition to information directly related to components, it is also proposed to add other types of information (e.g., floor plans and hazardous materials in the area) to available tags in the building. This allows facility users to access location-based data based on a predefined framework and could be used for emergency management and response.

Having multiple users who need to access different segments of the stored data requires solutions for data security and privacy. Data security is introduced using cryptography that ensures data confidentiality and integrity. Multi-level encryption together with role-based access control on the data stored on an RFID tag is proposed where a user is assigned a certain role and can only access the part of data that is authorized according to defined roles and an Access Control Policy (ACP).

2.1 Data capture method

In our proposed approach, every component is a potential target for tagging. Having tags attached to components would result in a massive tag cloud in the building. While having tags attached to all components would not happen in the immediate future, in order to benefit from the concept of having memory tags on a mass of objects, the subset of components to be tagged can be selected based on the scale of the project, types and values of the components, specific processes applied to these components, and the level of automation and management required by the facility owners.

The target components are tagged during or just after manufacturing and are scanned at several points in time. The scan attempts are both for reading the data, or modifying the data based on the system requirements and the stage at which the scan is happening. The scanned data are transferred to different software applications and processed to manage the activities related to the components. Moreover, software applications synchronize data on tags and BIM after each scan attempt.

Data modifications on tags are executed by different types of RFID readers (stationary or mobile). In order to identify the suitable type of readers for each scan attempt, the detailed process requirements should be captured, such as the readability range, data transfer rate and portability.

In the proposed approach, RFID tags are fixed to components; therefore, tags should be designed to have the maximum possible range and protection from noise and interference. However, it is always possible to control read/write range of the reader based on the process requirements.

2.2 Conceptual data structure

Considering the limited memory of the tags, the subset of BIM data stored on the tags has to be chosen based on the requirements. Because data on a tag are changing during the lifecycle of the
component and different software applications use and modify the data with different designated access levels, the memory of the tag should be virtually partitioned in a structured fashion based on predefined data types. We propose to virtually partition the memory space into the following fields:

**ID:** In order to look up the component in the BIM database, a none-changeable, unique identifier for each component is needed.

**Specifications:** This field is dedicated to specifications of the component derived from the design and manufacturing stage of the lifecycle. Safety related information and hazardous material information are examples of these specifications.

**Status:** Status field identifies the current main stage (e.g., in service, installed, and assembled) and sub-stage (e.g., in service: waiting for inspection) of the lifecycle. The status information is used to decide which software application can use and modify the data in the process data field.

**Process data:** This field is relatively large compared to the other fields and is designed to store the information related to the component’s current stage of the lifecycle. The data related to current processes to be stored on the tags are different and should be changed during the lifecycle. For example, assembling instructions are used only in the assembly stage. Therefore, the process data field contains only information related to the current lifecycle stage taken from the BIM database. Different applications use the same memory space but at different lifecycle stages.

**History data:** This field is designated for storing the history data used during the lifecycle for maintenance and repair purposes. The history records are derived from BIM and accumulated during the lifecycle to be used in forthcoming stages.

**Environment data:** This field is designated for storing environment specific data, such as the location of a component and the specifications of the space (e.g., floor plan). Environment data is also extracted from BIM and contains all the information that is not related to the component itself.

Figure 1 shows how the data stored on the tag can be used in the operation and maintenance phase by different users. During the normal operation of a component, information such as user manual, occupant names, map and locations, can be used by regular facility users. During fire incidents other types of stored data (such as hazardous materials in the area or evacuation procedure) can be used by...
fire fighters and other users. Furthermore, the data on the tag can be used by facilities management personnel to facilitate the process of inspection or maintenance.

2.3 Data encryption of RFID-based distributed storage

The data is stored on RFID tags in XML format. Thus, the goal is to derive a data protection scheme that can strictly enforce this XML ACP (XACP) and at the same time minimize the storage. Further, it is assumed that the XACP is defined in terms of Role-Based Access Control (Saini et al., 2009).

As shown in Figure 2, the procedure of data encryption consists of the following steps:

Access Tree Generation: The input is the original XML file, which should be protected while providing services to the authorized users, and the corresponding XACP. The basic idea is to compress a connected part of the original XML tree structure that is accessible by the same set of roles into a single node. The output is called an Access Tree.

Due to its hierarchical nature, an XML document can be represented as non-cyclic directed tree $T = (N, B)$ where $N$ is the set of all the nodes $N = \{ n_1, \ldots, n_n \}$ in $T$ and $B$ is the set of all the branches $B = \{ b_1, \ldots, b_m \}$ in $T$. In our context, the XACP is represented with a set of subject-object relations, where subject is a role and object is a set of elements within the XML document that are accessible to this role. Let $N = \{ R_1, \ldots, R_r \}$ denote the set of all the roles defined in the XACP. Let $\{ N_i \}$ denote a subject-object relation, where $R_i$ is a specific role in $R$ and $N_i = \{ n_i_1, \ldots, n_i_k \}$ is the set of nodes in $T$ that are accessible to users assigned with role $R_i$. Taking $T$ as the input, we mark each node of $T$ (denoted as $n$) with a set containing the roles that can access this node, which is called the Role Set of node $n$, and is denoted as $M(n)$.

![Figure 2. Process of XML encryption](image)

After all the nodes are marked, a search is performed from the root of $T$ for access tree generation. At each step of the search, the marking set of the current node $n$, i.e., $M_n$, is compared to that of the next node $n'$, i.e., $M_{n'}$. If $M_n = M_{n'}$, the branch between $n$ and $n'$ is removed, and $n'$ is merged into $n$. At the end of this depth-first-search, the access tree corresponding to $T$ is generated. An example is shown in Figure 3(a). The detailed access information about each role is shown in Table 1. Initially, considering only the first four roles, the step of access tree generation is shown in Figure 3(b).

Policy Type Checking: An algorithm is executed to determine the type of XACP. The ACP is called simple policy, if the role set of any node in the access tree is a subset of that of its parent node, if any. Otherwise, it is called a complex policy. Figure 3(c) shows that the role set of node D is not a subset of its parent node. Thus, this new XACP is complex.

Access Tree Transformation: the access tree transformation step is applied only when the XACP is identified as a complex one through the policy type checking step (such as the case when adding $R_5$). In general tree structure encryption, encrypting a node literally means encrypting only the content of that particular node. However, in XML encryption, encrypting a node $n_i$ means encrypting both $n_i$ and all its child nodes, i.e., $T_{n_i}$. As a result, in terms of a complex XACP, it is impossible to perform multi-level encryption according to the hierarchical structure of the access tree.

To address this issue, the original XML document has to be transformed in such a way that a simple XACP can be applied on the transformed document, which is equivalent to the complex XACP enforced on the original document. The transformation can be done through either copying or moving a certain part of the document to another position. In our applications, since we are more concerned about storage efficiency, Access Tree Transformation Algorithm (Saini et al., 2009), involves only
moving operations. In terms of the complex XACP example above, the resulting tree without transformation is shown in Figure 3(c) and the transformed access tree is shown in Figure 3(d).

**Role Key Generation and Multi-Level Encryption:** To reduce the key management overhead, it is desirable that each role maintain a minimum number of keys. In our scheme, through identifying complex policies and making appropriate transformations, Crampton’s scheme can be applied to the resulting access tree (Crampton, 2004), no matter which type of ACP needs to be enforced on the original XML document.

Taking the resulting access tree in Figure 3(d) as an example, each role $R_i$ needs to maintain only one key, i.e., $k_i$. Different parts of the document that are accessible to users with role $R_i$ can be decrypted by either $k_i$ or other keys derived from $k_i$. Finally, the XML document is encrypted according to the method proposed by Miklau and Suciu (2003) with these role keys. More details on the process of role key generation and encryption framework could be found in Saini et al. (2009).

![Figure 3. Access tree generation/transformation](image)

**3 Case Study**

In this case study, RFID tags are used for storing information about fire extinguishers at Concordia University. Fire extinguishers should be regularly inspected, maintained, recharged and tested. Strict safety regulations force the owners to spend large amount of money to perform inspection on a regular basis. In our prototype system, crucial information is stored on tags attached to the extinguishers and is updated by various users such as inspectors and facilities management staff.

Active tags with 8 or 32 KB of memory and relatively long range that work well near liquids and metal have been used to store information. Short write distance for tags would guaranty that the inspector did the inspection and maintenance activity in close proximity of extinguishers.

The memory of the tags has been segmented following the general data structure introduced in Subsection 2.2. The data are saved in XML format on the tags memory. A predefined ACP contains a set of roles and access rules to different parts of XML data. By applying the encryption method introduced in Subsection 2.3, the data is encrypted and saved on the tags. Hence, unauthorized users can not interpret and maliciously use the data.

The application extends the work in a previous facilities management project (Motamedi and Hammad, 2009), by adding different roles and security levels. Easy identification of components is also achieved by developing a user interface that shows scanned information from RFID tags on the floor plan. Figure 4 shows a sample snapshot of the screen, where the locations of sensed RFID Tags
are shown with green stars and the selected RFID tag is shown with a red square shape on the floor plan to help the user visually identify the components on the plan.

After finding the extinguisher, the inspector starts the tasks by reading the data on the extinguisher tag. The software shows decrypted data specific to the selected role. The data helps the inspector to quickly review the history of inspection, previous maintenance results and information about the type and possible defects of the extinguisher. Moreover, the GUI provides the users with related alerts or guidelines based on the selected role and the recorded data. For example, the software informs the inspector about inspection procedures specific to the scanned extinguisher. The inspector performs the tasks and completes easy-to-fill forms, which is faster and more accurate than conventional paper-based systems. The inspector views the result of inspection/maintenance activity and confirms. Before the software updates the data on the tag, it encrypts the modified data with the inspector role key. The task is considered to be completed only after the successful data update of RFID.

**Figure 4. Software snapshot**

4 **Conclusions and Future Work**

The paper provides conceptual data structure, data encryption procedure and implementation approach of a futuristic vision of facilities with RFID tags attached to their components. Several challenges should be addressed to make the vision practical and financially feasible. The following steps are necessary for realizing the proposed approach: (1) identifying most suitable components for tagging based on cost-benefit analysis, (2) re-engineering existing processes for selected components, (3) investigating product-specific tag data for each lifecycle stage, (5) technology selection and field testing for RFID hardware, and (6) investigating RFID-related information to be added to BIM.

**References**


