Winter 12-5-2004

Security Service Model for RFID Enabled Supply Chain

Bo Chen

Hongying Gu

Oliver Wang

Security Service Model for RFID Enabled Supply Chain

Bo Chen[1], Hongying Gu[2], Oliver Wang[3]

[1] Software College, Zhejiang University of Technology, Hangzhou 310032, China
[2] Computer College, Zhejiang University, Hangzhou 310027, China
[3] SAP China Labs, Shanghai 200003 China
[1] cb@zjut.edu.cn, [2] guhy@zju.edu.cn, [3] Oliver.wang@sap.com

ABSTRACT

It has been widely recognized that RFID related technologies will greatly improve the visibility, the efficiency and the collaboration of industry supply chain. In this new "product driven" supply chain scenario, manufacturer, supplier, and third party share and coordinate the use of diverse resources in distributed "virtual organizations". It challenges the security issues, which demand new technical approaches. In collaboration with researchers in Auto-ID Labs China, we have developed a service-oriented framework based on CA and Web Services, which supports inexpensive mediation of product information among rapidly evolving, heterogeneous information sources. Our architecture defines a user-driven security model that allows users to create entities and policy domains within virtual organizations. We emphasize that standard Web Services tools and software provide both stateless and stateful forms of secured communication.

Keywords: security service, RFID, supply chain, PKI

1. INTRODUCTION

Radio frequency identification (RFID) is a powerful new technology that is helping companies to improve visibility into their inventory data and increase accuracy across supply chain networks. When RFID tags are attached to products, boxes, and pallets, the items can be followed automatically as they move, providing a real-time and accurate view of inventory[1]. The technology is so effective that organizations such as Wal-Matt, Metro, and the U.S. Department of Defense now require their suppliers to deliver goods with RFID tags.

The Auto-ID Center in MIT, was launched in late 1999, has developed RFID tags which are cheap enough to be disposable and readers which are agile enough to de deployed globally[2]. According to Auto-ID Center's researching, the various components of the Auto-ID infrastructure includes (Figure 1):

- Hardware – such as RFID tags and readers but also including optical readers and other types of sensors.
- Savant – a software platform for interfacing to various types of readers and filtering the data generated when the tags are within range of the readers.
- The Electronic Product Code (EPC) – a unique identification number for each object – the minimum information which must be stored on its ID tag.
- The Object name service (ONS) – an extension of the domain name service (DNS), which provides a method of converting an EPC into the network address of the corresponding networked database, which in turn holds the product data for that object.
- The EPC Information Server (formerly PML Server) – an open standard XML Schema for representing information about products and communicating this information globally in a unified manner, between parties who otherwise have disparate databases, data storage formats and who use diverse application programs to act upon the product data.
- Control systems – to provide feedback into the supply chain or manufacturing process, using the data generated by EPC readings.

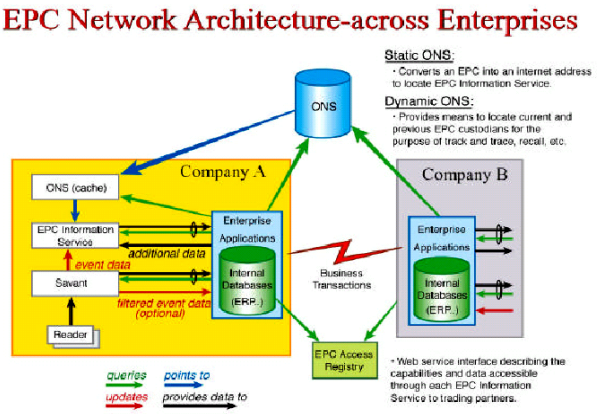

Figure 1 EPC Network Architecture
Source: Autoidcenter.org

Since each unique object will have a unique identifier, in the form of the EPC, each object will become an individual, with its own individual history about how it passed the through supply chain. In this "product driven" supply chain scenario, manufacturer, supplier, and third party (like carrier, distribute center) share and coordinate the use of diverse resources in distributed "virtual organizations".

Let us consider a simple business scenario. The suppliers and customers (manufacturers) collaborate by sharing information and exchanging plans via the Supplier Managed Inventory (SMI) process. The SMI process is a supplier-driver replenishment and planning process, which is based on min/max stock balance levels, on current demand, in-transits and planned shipments.

With RFID enabled, the application now should support the handlingof massive amounts of Auto-ID data through connectivity with readers, tags, and other devices; and integration of high-volumn RFID data with back-end business processes. The dynamic and multi-parties nature of these environments introduces challenging security issues that demand new technical approaches. Most of data are sensitive, and every server exposed on internet will be attacked, and so on.

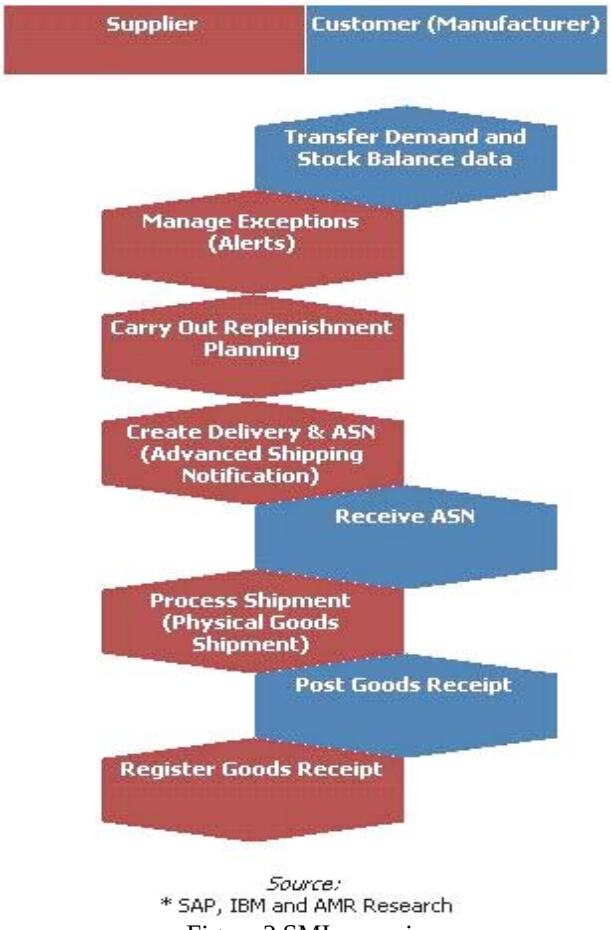

Figure 2 SMI scenario

The remainder of this article is organized as follows. In Section 2, we investigated related work on security web service and certificate validation. Then we propose a design of security system model, emphases on security web service and privilege management infrastructure. Finally, we explain function for system and then we conclude this paper with a brief discussion of future work.

2. RELATED WORK

2.1 CA-based Trust Model

One commonly used identity token is provided by an X.509 public key certificate. An X.509 “public key certificate” is defined in ITUT standard [3] as “the public key of a user, together with some other information, rendered unforgeable by encipherment with the private key of the certification authority which issued it.” [Section 3.3.44]

In practice an X.509 certificate as defined by the PKIX X.509 Certificate Profile RFC[] contains a public key, a subject name in the form of a multi-component Distinguished Name [DN] and a validity period and is signed by a trusted third party called a Certification Authority (CA). The X.509 certificates are used with the TLS(SSL) security protocol to make a secure authenticated connection between two parties. X.509 certificates are exchanged between entities, usually in a phase where no security has yet been negotiated (no data integrity or confidentiality). The certificates are first tested for validity by checking the expiration dates, possible revocation, acceptable key usage and if they are signed by trusted CA. If the certificates pass all these checks their public keys are then used to build a challenge handshake to prove that each entity that sent a certificate has the corresponding private key.

There are other public/private key based schemes such as PGP keys[4], SSH keys [5]and the SPKI [6] keys and protocols. The PKIX/X.509 scheme has a smallish set of trusted third parties (CAs) to sign identity certificates that contain a subscriber’s public key. This improves the scaling properties of public key distribution since only the CA’s public key needs to be distributed in an out-of-band secure manner. Another feature of the X.509-TLS infrastructure is that it supports multiple independent CAs. In a supply chain, each company may choose which CAs it will accept for binding DNs and public keys. At this point, most of the current solutions allow for short-term proxy certificates, stored with unencrypted private keys, to which a user has delegated his identity. These certificates are correctly formatted X.509 certificates, except that they are marked as proxy certificates and are signed by an end entity rather than a CA[7].

2.2 XML Web Service Security

The XML security standards define XML vocabularies and processing rules. These standards use legacy cryptographic and security technologies, as well as emerging XML technologies, to provide a flexible, extensible and practical solution toward meeting security requirements. The XML security standards include XML digital signature [8]for integrity and signing solutions, XML encryption [9] for confidentiality, XML key management (XKMS) [10] for public key registration,

location and validation, security assertion markup languages (SAML) [11] for conveying authentication, authorization and attribute assertions, XML access control markup language (XACML) [12]for defining access control rules, and platform for privacy preferences (P3P) [13]for defining privacy policies and preferences. Major use cases include securing web services (WS-Security) [14].

2.3 RBAC Policies in Privilege Management

Role Based Access Controls (RBAC) has generated significant interest in the last decade. The main entities in RBAC are the users, the roles and the permissions. A role can represent a job function, a qualification or expertise. A permission represents the right to access a target in a particular mode. Roles are assigned to users in a many to many relationship, and permissions are granted to roles, again in a many to many relationship. There is thus a level of indirection between a user and his access rights. The benefits of RBAC are widely discussed on lots of papers [15].

Edition 4 of X.509 [3], published in 2001, is the first edition to fully standards the certificates of a Privilege Management Infrastructure (PMI). X.509 supports RBAC by defining role specification attribute certificates that hold the permissions granted to each role, and role assignment attribute certificates that assign various roles to the users. Each role and name component in X.500 and LDAP [16] is an attribute type, attribute value pair. Thus roles and name components are easily interchangeable. The user is identified by either his/her LDAP Distinguished Name or his public key certificate (issuer and serial number). Role assignment ACs may point to their corresponding role specification AC via the role specification certificate identifier extension.

A policy is defined which states the rules for assigning roles to users, and permissions to roles. The policy can then be used control the accesses to all the targets within the policy domain [17]. In practice, now we have APIs that reads in the XML Policy, parses it, and the uses it to control access to targets within the policy domain. The API caller, typically an application gateway, passes the authenticated name of the user, and this is used to retrieve the user's attribute certificates (ACs) from the configured directory (In most case, LDAP server is used). The API caller passes the user's requested action on his chosen target, and again this is checked against the policy. The API returns either granted or denied to the caller.

3. SECURITY MODEL FOR RFID-ENABLED SCM

We now turn to outline the proposed structure of a prototype SSS (SCM Security Server).

Web services technologies allow software components to be defined in terms of access methods, bindings of these methods to specific communication mechanisms, and mechanisms for discovering relevant services. Now, the Simple Object Access Protocol (SOAP) and the Web Services Description Language (WSDL) are widely used in many applications.

SSS defines standard Web service interfaces and behaviors that add to Web services the concepts of stateful service and secure invocation, as well as other capabilities needed to address RFID-specific requirements. These interfaces and behaviors define what is called a "SSS Service" and allow users to manage the web service's life-cycle, as allowed by policy, and to create sophisticated distributed services.

Our SSS security model casts security functions as SSS services. This strategy allows well-defined protocols and interfaces to be defined for these services and permits an application to outsource security functionality by using a security service with a particular implementation t fit it current need.

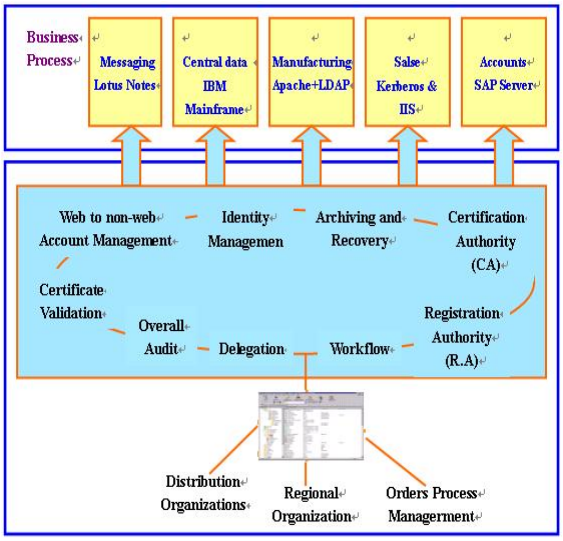

Figure 3 SSS for RFID enabled SCM

The security services include the following.

- Certificate Validation service (CAS): A service that handles the details of processing and validating the binding among the subject identity, the subject public key, and subject attributes.
- Authorization service: A service that evaluates policy rules regarding the decision to allow the attempted actins based on information about the requestor, and details of the request.
- Identity Management service: A service that takes a user's identity in one domain and returns the identity in another (e.g between X.509 and Kerberos).
- Archiving and Recovery: A service that archiving and recovery of encryption keys.
- Administration Service: provide administration workflow with approval and escalation controls.
- Overall Audit: A service that securely logs relevant information about events.

For example, SSS perform path validation on a certificate chain according to the local policy and with local PKI facilities, such as certificate revocation (CRLs) or through an OCSP(Online Certificate Status Protocol). The SSS client generates an 'SSS validate' request. This is essentially asking the SSS server to go and find out the status of the server's certificate. Then the SSS server performs a series of validation tasks. After certificate status is determined, SSS server replies to client application with status of the server's certificate and application acts accordingly. We use CA database connection protocol for the purpose of that the server obtains real-time certificate status information from CAs. The client use OCSP protocol.

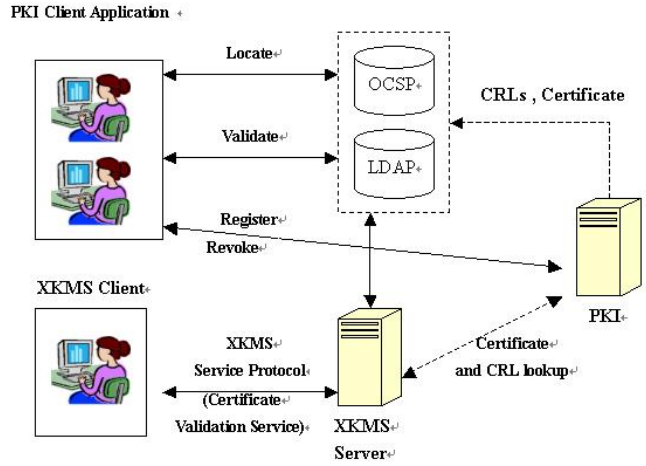

Figure 4 Identify Service

4. PMI IMPLEMENTATION

There are three main components to the PMI implementation – the authorization policy, the privilege allocator (PA) and the PMI application programming interface (API).

4.1 Authorization Policy

The authorization policy specifies who has what type of access to which targets and under what conditions. The xml.org published a document type definition (DTD) for the X.500 PMI RBAC policy. It provided a meta-language that holds the rules for creating the XML policies. The DTD comprise the following components.

- SubjectPolicy
- RoleHierarchyPolicy
- SOAPolicy
- RoleAssignmentPolicy
- TargetPolicy
- ActionPolicy
- TargetAccessPolicy

The SOA creates the authorization policy for the domain using his or her preferred XML editing tool and stores this in a local file to be used later by the PA.

4.2 Privilege Allocator

The SOA uses the PA to allocate roles to users in the form of role-assignment ACs. In the simple case of RFID supply chain, there are two roles – ASN issuer and ASN receiver. Issues can download any ASN that are produced.

Once the role-assignment ACs have been created by the PA, they are stored in a LDAP directory.

Another function of the PA is to create an authorization policy that is signed digitally as a policy AC. The policy AC is a standard X.509 AC. The PA prompts the SOA for the name of the policy fiel and then it copies the contents into the attribute value. After the SOA has signed the policy AC, the PA stores it in the SOA's entry in the LDAP directory.

4.3 API

A standard authorization API has already been defined by the Open Group in 2000. It is called the AZN API and is specified in the C language. It is based on the ISO 10181-3 access control framework and specifies the interface between the access control enforcement function (AEF) and the access control decision function (ADF).

We rewrite the AZN API with Object-Oriented method, and select Java language for development. After these works, the application gateway use API to "construct" and PMI Java Object, and complete the work by invoke the methods of this object. The main methods include: GetCreds, Decision, and Shutdown.

When a user initiates a call to the target, the AEF authenticates the users, then passes LDAP DN to the ADF through a call to GetCreds. The ADF uses this DN to retrieve all role ACs of the user from the list of LDAP URIs ("pull mode"). The role ACs are validated against the policy e.g. to check that the DN is within a valid subject domain, and to check that the ACs are within thje validity time of the policy etc. Once the user has been successfully authenticated he will attempt to perform actions on the target. At each attempt, the AEF passes the subject object, the target name, and the attempted action along with its parameters, to the ADF via a call to Decision. The Decision method will return "Granted" or "Denied" after executing the checking.

Shutdown method is designed to used in dynamically deploy new policy in domain. The AEF can follow the call to Shutdown with a new Constructor call, and this will cause the ADF to read in the latest policy and be ready to make access control decisions again.

5. EXPERIMENT

The first demo system for SSS has been implemented based on the design described in previous section. Some

open source projects are used in the demo system, like Tomcat, Apache SOAP, openCA, and so on. The security API depends core Java security technology, includes Java Authentication and Authorization Service (JAAS), Java Cryptography Extensions (JCE) and Java Secure Socket Extensions (JSSE).

Security service is complicated system. From the running result, there are many works to do in the future. We should consider the performance on the real Internet environment. We need add more robust integration with exist system. And next step, the private keys are generated in the Tags or cards, they can be used as soon as the user is in possession of his tag or card.

6. CONCLUSION

RFID technology is changing the way organizations manage their supply chain networks. With RFID, inventory data becomes more transparent and more accurate, enabling organizations to speed up delivery, respond more quickly to customer demand. But integrating RFID into existing systems has been prolematic, especially, it let suppliers, manufacturers, and customers become a virtual organization across the unsafe internet. We propose a security approach on open web service to validate certificate based on current web security environment, and a privilege management infrastructure used in collaborative business scenario Our approach will be a model for the future security system that offers security of open web service security.

ACKNOWLEDGEMENT

The author would like to thank AutoID Labs China. This work was supported in part by Ubipass company.

REFERENCES

[1] SAP RFID Home. http://www.sap.com/solutions/scm/rfid/.
[2] Auto-ID Labs. http://www.autoidlabs.org.
[3] ITU-T Recommendation X.509 | ISO/IED 9594-8, “The Directory: Publib-Key and Attribute Certificate Frameworks”, Draft V4 Feb 23,2001.
[4] S.Garfinkel: PGP:Pretty Good Privacy. O’Reilly & Associates, 1994.
[5] D. Barret, R. Silverman, SSH: the Secure Shell, O’Reilly & Associates, 2001.
[6] C. Ellison, SPKI Requirements, IETF RFC 2692, 1999.
[7] Mary R. Thompson etc., “CA-based Trust Model for Grid Authentication and Identity Delegation”, Oct 2002.
[8] XML-Signature Syntax and Processing (W3C/IETF Recommendation), Feburary-2002
[9] XML Encryption Syntax and Processing (W3C Recommendation), 2003
[10] XML Key Management Specification Version 2.0 (W3C Working Draft), April 2003.
[11] Security Assertion markup language. http://www.oasis-open.org
[12] eXtensible Access Control markup language, http://www.oasis-open.org
[13] Platform for privacy preferences, http://www.w3.org/P3P/
[14] Web Services Security (WS-Security), http://www-106.ibm.com/developerworks/webservices/library/ws-secure/
[15] ACM Workshop on Role Based Access Control, 1996-2001. see http://portal.acm.org/portal.cfm for proceedings.
[16] Lightweight Directory Access Protocol, RFC 2251, Dec. 1997.
[17] B.Moore, etc. , “Policy Core Information Model”, RFC 3060, Feb 2001.