Trusted Certificate Validation Scheme for Open LBS Application Based on XML Web Services

Kiyoung Moon*, Namje Park*, Kyoil Chung*, Sungwon Sohn*, and Jaecheol Ryou**

Abstract: Location-based services or LBS refer to value-added service by processing information utilizing mobile user location. With the rapidly increasing wireless Internet subscribers and world LBS market, the various location based applications are introduced such as buddy finder, proximity and security services. As the killer application of the wireless Internet, the LBS have reconsidered technology about location determination technology, LBS middleware server for various application, and diverse contents processing technology. However, there are fears that this new wealth of personal location information will lead to new security risks, to the invasion of the privacy of people and organizations. This paper describes a novel security approach on open LBS service to validate certificate based on current LBS platform environment using XKMS (XML Key Management Specification) and SAML (Security Assertion Markup Language), XACML (extensible Access Control Markup Language) in XML security mechanism.

Keywords: Location-based service, Open LBS security, XKMS, XML security, XML web services

1. Introduction

Recently, with the rapid development of mobile communication technology and wide spread of mobile devices such as cellular phones equipped with a GPS (Global Positioning System) receiver, PDA (Personal Digital Assistant), notebook PCs, location-based services (LBS) technology which uses location information of mobile devices is being more important. In LBS, a LBS platform which stores, manages location information of mobile devices, and provides common and basic functions for location based applications is very important. This new technology is giving a great impact to how we live and do businesses. Knowing the physical position of a user at any given time can be a huge potential to application service providers. This service allows mobile users use services based on their position or geographic location.

LBS platform should provide fundamental functions such as the acquisition of location information, security, user privacy management, authentication, and management of a large volume of location data. To be successful, an LBS technology has to provide an accurate location, as well as suitable information for users required by the corresponding service, with minimal expenditure including establishing infrastructure and overhead. The most important technology is of course the positioning technology, the way to find out the location of a mobile device accurately. Due to the unique characteristics of the cellular environment, it is a great challenge to locate the object precisely. Many advanced methods are used for positioning.

LBS requests can span multiple security domains. Trust relationships among these domains play an important role in the outcome of such end-to-end traversals. A service needs to make its access requirements available to interested client entities, so that they understand how to securely request access to it. Trust between end points can be presumed, based on topological assumptions or explicit, specified as policies and enforced through exchange of some trust-forming credentials. In a LBS environment, presumed trust is rarely feasible due to the dynamic and distributed nature of virtual organizations relationships. Trust establishment may be a one-time activity per session or it may be evaluated dynamically on every request. The dynamic nature of the LBS in some cases can make it impossible to establish trust relationships among sites prior to application execution. Given that the participating domains may have different security infrastructures it is necessary to realize the required trust relationships through some form of federation among the security mechanisms.

Furthermore, open LBS service infrastructure will extend use of the LBS technology or services up to business area using web service technology. Therefore differential resource access is a necessary operation for users to share their resources securely and willingly. Therefore, this paper describes a novel security approach on open LBS service to validate certificate based on current LBS environment using XKMS (XML Key Management Specification) and SAML (Security Assertion

Manuscript received October 4, 2005; accepted November 8, 2005.
The fifth author of this research was supported by University IT Research Center Project of Korea MIC (Ministry of Information and Communication.)

* Information Security Research Division, ETRI, Daejeon, Korea ({kymoon, namejepark, kyoil, swsohn}@etri.re.kr)
** Department of Computer Science, Chungnam University, Daejeon, Korea (jcryou@home.cnu.ac.kr)

Markup Language), XACML (eXtensible Access Control Markup Language) in XML (eXtensible Markup Language) security mechanism.

This paper is organized as follows. First we investigate related work on LBS and mobile web service, XML web service security. Then we propose a design of security system platform for open LBS service and explain experimented XKMS model for certificate validation service. Finally, we explain function of system and then we conclude this paper.

2. Mobile XML Web Services

A mobile XML web service can feature one of the following architectures: wireless portal network, wireless extended Internet, or wireless ad hoc network.

In a wireless portal network, the wireless information devices connect to the Internet backend services through portal entry points. The portal creates a "walled garden" and controls access to Internet contents. Wireless portal networks support widely deployed thin-client wireless technology, such as WAP (Wireless Application Protocol). The portal receives the message, checks the user's privilege, and then translates the request to a SOAP (Simple Object Access Protocol) message or an XML-RPC call to an appropriate partner web service. The web service replies and the portal translate the response back to a WML (Wireless Markup Language) document. The portal sends the WML document back to the wireless device for display. In this way, the portal works as a proxy for wireless users. The portal operator provides user authorization and management services. Many partner vendors can provide real application web services under the ASP (Application Service Provider) model.

Wireless extended Internet is the wired Internet's expansion to wireless devices. Wireless information devices can have their own IP addresses (through Internet Protocol 6) and full network functionalities. Those devices usually run smart, fat clients that interact with multiple backend services simultaneously and store/process application data on the device. Smart devices support sophisticated user interfaces, offline processing, and automatic transactions. They can also implement flexible, application-specific security policies. Like the Internet itself, the wireless extended Internet architecture is decentralized and eliminates any single point of failure. However, as you will see later, centralized web services hubs are still required to support advanced security schemes and user interfaces. Unlike the portal architecture, the hubs themselves can be decentralized. Different vendors can provide similar hub services that can interoperate with each other. Fig. 1 shows a topography diagram for such networks.

The extended wireless Internet architectures blended with decentralized hub web services will provide the foundation for future wireless web services applications, an approach we focus on throughout this article. Since most of the supporting technologies are just emerging, many challenges prevail.

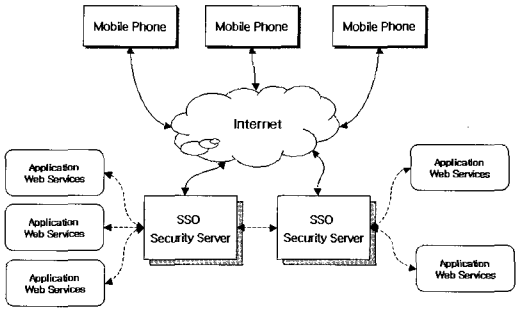

Fig. 1. Mobile Web Services Architecture.

The wireless ad hoc networks allow wireless devices to become servers to peers. Wireless peers can provide content, network traffic routing, and many other services. The ad hoc network truly leverages wireless networks' dynamic nature. However, because wireless peer-to-peer technology is still embryonic, its many performance and security issues must be solved before it can be widely used.

3. Security Considerations of Open LBS

3.1 Open LBS Architecture on the Basis of XML Web Services

In general, web services architecture is composed by a collection of services that are organized according to any functional aspects. An example of this kind of distributed architectures is the LBS frameworks, which integrate GIS and location services. According to the functional aspects related to this LBS context, a conceptual model of architecture has been defined. The presented architectural model has been the conceptual base for the development of a LBS framework whose functionality may be integrated into end applications through Internet (see fig. 3). Required services are organized according to the proposed functional levels and built according to the Web Service philosophy: their operations are provided through a standard, published interface to ensure interoperability, and are accessible via ubiquitous Internet protocols and data formats, such as HTTP and XML.

As the kernel of LBS system, the LBS platform taking charge of major roles for location services such as request/response for location information, location management, profile management, provisioning and authentication, roaming and network management. With the required functionality, the LBS platform ensures the interoperability with a wireless networks, mobile handset platforms and various contents. The LBS platform should be designed to open architecture presented in fig. 2 to complete the requirements.

The major roles of the LBS platform are as follows. First Interface functions to wireless network. The interface to GMLC (Gateway Mobile Location Center) / MPC (Mobile

Positioning Center) for acquiring the location information and the interface to wireless IP platform for portal service are needed. Second, the LBS platform provides a common API (Application Programming Interface) interface for supporting various location-based applications. The promising API is XML based common interfaces. And the last is the provision of application server. The LBS platform could include the application server in a broad sense, but the application server need a huge amount of memory, so the application server is departed from LBS platform in general.

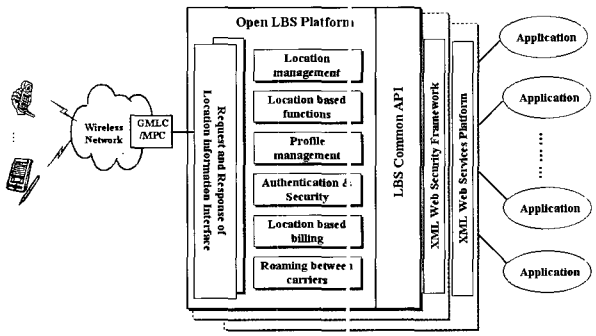

Fig. 2. Open LBS Web Service Architecture.

3.2 Security Considerations for Open LBS Service

As mentioned earlier, it is important to address information security concerns for open LBS service, especially for those applications that have anything to do with mobile e-business. The security aspect is a critical factor to realize the opportunities presented by L-Commerce (LBS mobile commerce) and the ubiquity of mobile devices. Mobile e-business exposes companies to a massive range of new threats and vulnerabilities.

The overall security of a LBS application is only as strong as its weakest link, and in an L-Commerce network, the weakest link is the mobile device [22]. The interceptable nature of wireless signals and the limited memory and computing power of most mobile devices leave wireless systems vulnerable to data theft. Some of the key security issues with LBS application security are as follows.

- Confidentiality : Access to confidential and sensitive data should be restricted to only those users who need it.
- Integrity : The integrity of data transmitted over wireless networks from the point of transmission to the point of delivery need to be extremely well maintained.
- Availability : Mission-critical data and services should be available with contingency plans to handle catastrophic events such as infrastructure failures, security breaches, etc.
- Privacy : LBS application developers should take care to adhere to the legal requirements to safeguard user privacy. This is particularly significant for location-based services as there is an inherent possibility that the users can be tracked. However, as we shall soon see, the availability of location information can be turned into a security advantage.

Effective wireless application security depends on the ability to authenticate users and grant access accordingly. Existing authentication and authorization mechanisms fundamentally depend on information known to a user (password or keys), possession of an authentication device (access token or crypto card) or information derived from unique personal characteristics (biometrics). None of this is totally foolproof.

Location information (latitude, longitude, height etc.) of a mobile device or user adds a fourth and new dimension to wireless application security. It gives extra assurance to users of the wireless applications who want to perform sensitive operations such as financial transactions, access valuable information, or remotely control critical systems. It can supplement or complement existing security mechanisms. Location information can still be used as a security mechanism when other systems have been compromised. For highly sensitive wireless applications, a broad geographic region could be specified as the set of authorized locations so that authorities can trace any malicious activity back to the location of the intruder. Without the incorporation of location information, it will be difficult to trace the origin of malicious activity.

Location information can provide evidence to absolve innocent users. If illegal activity is conducted from a particular user account by someone who has gained unauthorized access to that account, then the legitimate owners of the account might be able to prove that they could not have been present in the location where the activity originated. Location-based authentication can be done transparently to the user and be performed continuously. This means that unlike most other types of authentication information, location information can be used as a common authenticator for all systems the user accesses.

In summary, adding location-based authentication to security mechanisms helps restrict access to sensitive information or transactions, prevent unauthorized access, track down illegal activity, and possibly ensure that only devices positioned within a certain area can receive encrypted information.

4. Design of Security Framework for Providing Secure Open LBS

4.1 Security Mechanism for Open LBS Environment

The XML security standards define XML vocabularies and processing rules in order to meet security requirements. 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. XML web security services can be used to provide LBS security service by standardizing and integrating leading security solutions (such as Kerberos authentication and authorization, digital certificates, digital signatures, and public/private key encryption) using XML messaging [1,2,3,8]. XML messag-ing is referred to as the leading choice for a wireless communication protocol and there are security protocols for LBS applications based upon it. Among them are the following.

- SAML, which is a protocol to transport authentication and authorization information in an XML message. It could be used to provide single sign-on web services.
- XML signatures define how to digitally sign part or all of an XML document to guarantee data integrity. The public key distributed with XML signatures can be wrapped in XKMS formats.
- XML encryption allows applications to encrypt part or all of an XML document using references to pre-agreed symmetric keys.
- The web services secure XML protocol family (WS-Security), endorsed by IBM and Microsoft, are a complete solution to provide security to Web services. It is based on XML signatures, XML encryption, and an authentication and authorization scheme similar to SAML.

All of the above security protocols can bind to web services messaging protocols [9,17]. For example, we can embed a SAML segment in a SOAP message header to authenticate and authorize the access to the requested services. We can also embed an XML signature segment in a SOAP header to authenticate a credit card number in that message. XML security mechanisms raise the possibility of a new architecture where security plays a key role.

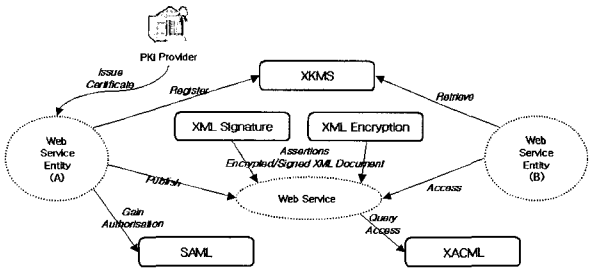

Fig. 3. Interworking Security XML Web Services.

Fig. 4 illustrates the layering of existing security technologies and standards and shows how these fit into the LBS security model. Moving from the machine and OS security on the bottom to the applications and server environment at the top, one can identify different layers that either are built and depend on their lower neighbors, or are a level up in abstraction.

The same or similar functions can be implemented at different levels, with different characteristics and tradeoffs. For example, security can be an inherent part of a network and binding layer. In the case of the network layer, it can be provided via IPSec (Internet Protocol Security) or SSL (Secure Socket Layer) / TLS (Transport Layer Security). In the case of the binding layer, it can be provided by HTTPS and in the case of IIOP (Internet Inter-ORB Protocol), by CSIv2 (Convergence sublayer identifier). In a messaging environment, the message provider (e.g., MQ) can provide end-to-end message security. Given the increasing use of XML, the security standards in the XML space play an important role here: XML signature, XML encryption, XML key management service (XKMS), and assertion languages (e.g., SAML). Built on top of XML standards are the web services standards, including WSDL (Web Services Description Language).

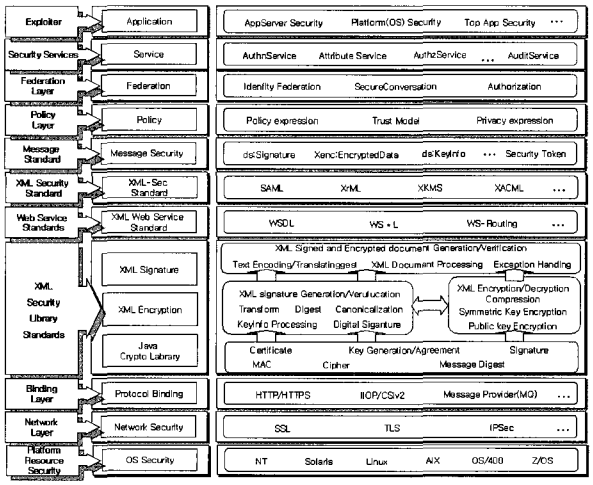

Fig. 4. Security Layer blocks for Open LBS based on XML Security Mechanism.

4.2 Design of Open LBS Security Service Framework

Web services can be used to provide mobile security solutions by standardizing and integrating leading security solutions using XML messaging. XML messaging is referred to as the leading choice for a wireless communication protocol and there are security protocols for mobile applications based upon it. Among them are the follows. SAML is a protocol to transport authentication and authorization information in an XML message. It could be used to provide single sign on web services. XML signatures define how to digitally sign part or all of an XML document to guarantee data integrity. The public key distributed with XML signatures can be wrapped in XKMS formats. XML encryption allows applications to encrypt part or all of an XML document using references to pre-agreed symmetric keys. The WS-Security, endorsed by IBM and Microsoft, is a complete solution to provide security to web services. It is based on XML signatures, XML encryption, and an authentication and authorization scheme similar to SAML. When a mobile device client requests access to a back-end application, it sends authentication information to the issuing authority. The issuing authority can then send a positive or negative

authentication assertion depending upon the credentials presented by the mobile device client. While the user still has a session with the mobile applications, the issuing authority can use the earlier reference to send an authentication assertion stating that the user was, in fact, authenticated by a particular method at a specific time. As mentioned earlier, location-based authentication can be done at regular time intervals, which means that the issuing authority gives out location-based assertions periodically as long as the user credentials make for a positive authentication.

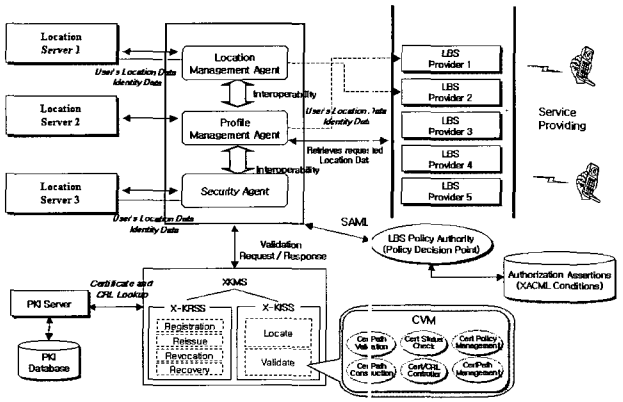

Fig. 5. Security Service Model for Open LBS.

CVM (Certificate Validation Module) in XKMS system perform path validation on a certificate chain according to the local policy and with local PKI (Public Key Infrastructure) facilities, such as certificate revocation (CRLs) or through an OCSP (Online Certificates Status Protocol). In the CVM, a number of protocols (OCSP, SCVP, and LDAP) are used for the service of certificate validation. For processing the XML client request, certificate validation service from OCSP, LDAP (Lightweight Directory Access Protocol), SCVP (Simple Certificate Validation Protocol) protocols in XKMS based on PKI are used. The XKMS client generates an 'XKMS validate' request. This is essentially asking the XKMS server to go and find out the status of the server's certificate. The XKMS server receives this request and performs a series of validation tasks e.g. X.509 certificate path validation. Certificate status is determined. XKMS server replies to client application with status of the server's certificate and application acts accordingly. Using the OCSP protocol, the CVM obtained certificate status information from other OCSP responders or other CVMs. Using the LDAP protocol, the CVM fetched CRL (Certificate Revocation List) from the repository. And CA (Certificate Authority) database conne-ction protocol (CVMP;CVM Protocol) is used for the purpose of that the server obtains real-time certificate status information from CAs. The client uses OCSP and SCVP. With XKMS, all of these functions are performed by the XKMS server component. Thus, there is no need for LDAP, OCSP and other registration functionality in the client application itself.

5. Open LBS Security Services Invocation Process

5.1 Invocation Process of Security protocol

Three types of principals are involved in our protocol: LBS application (server/client), SAML processor, and XKMS server (including PKI). Proposed invocation process for secure LBS security service consists of two parts: initialization protocol and invocation protocol. The initialization protocol is prerequisite for invoking LBS web services securely. Through the initialization protocol, all principals in our protocol set up security environments for their web services, as shown in fig. 6. The flow of setting up security environments is as follows.

The client first registers its information for using web services, and then gets its id/password that will be used for verifying its identity when it calls web services via secure channel. Then, the client gets SAML assertions and installs security module to configure its security environments and to make a secure SOAP message. It then generates a key pair for digital signature, and registers its public key to a CA.

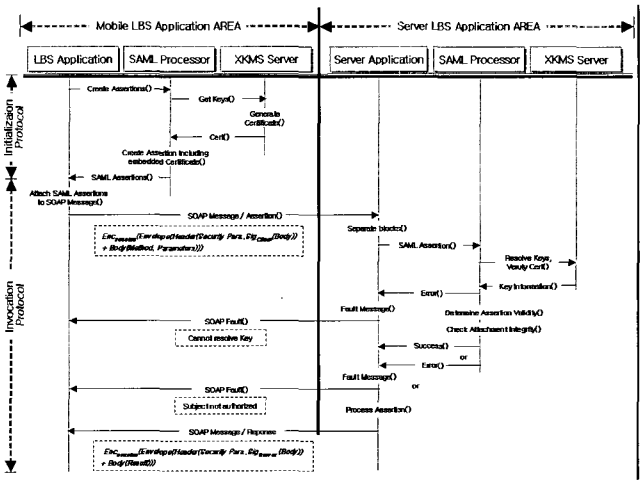

Fig. 6. Security Protocol for Secure Open LBS Service.

The client creates a SOAP message, containing authentication information, method information, and XML signature, XML encrypts it, and then sends it to a server. The message is in following form: $Enc_{session}$*(Envelope (Header (SecurityParameters,* Sig_{client}*(Body)) + Body(Method, Parameters))))*, where $Sig_x(y)$ denotes the result of applying x' s private key function (that is, the signature generation function) to y. The protocol shown in fig. 6 shows the use of end-to-end bulk encryption [25]. The security handlers in server receive the message, decrypt it, and translate it by referencing security parameters in the SOAP header. To verify the validity of the SOAP message and authenticity of the client, the server first examines the validity of the client's public key using XKMS. If the public key is valid, the server receives it from CA and verifies the signature. The server invokes web services after completion of examining the security of the SOAP

message. It creates a SOAP message, which contains result, signature, and other security parameters. Then, it encrypts the message using a session key and sends it back to the client. Lastly, the client examines the validity of the SOAP message and server, and then receives the result.

In current LBS service, there is no mechanism of differential resource access. To establish such a security system we are seeking, a standardized policy mechanism is required. We employ the XACML specification to establish the resource policy mechanism that assigns differential policy to each resource (or service). SAML also has the policy mechanism while XACML provides very flexible policy mechanism enough to apply to any resource type. For our implementing model, SAML provides a standard-ized method to exchange the authentication and authori-zation information securely by creating assertions from output of XKMS (e.g. assertion validation service in XKMS). XACML replaces the policy part of SAML as shown in fig 7.

Once the three assertions are created and sent to the protected resource, there is no more verification of the authentication and authorization at the visiting site. This, SSO (Single Sign-On), is a main contribution of SAML in distributed security systems.

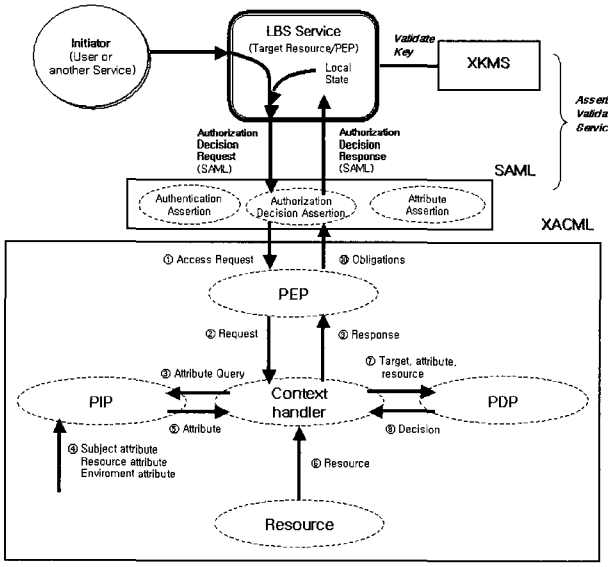

Fig. 7. SAML/XACML Message Flow using XKMS in Open LBS.

Fig. 7 shows the flow of SAML and XACML integration for differential resource access. Once assertions are done from secure identification of the PKI trusted service, send the access request to the policy enforcement point (PEP) server (or agent) and send to the context handler. Context handler parses the attribute query and sends it to PIP (policy information point) agent. The PIP gathers subject, resource and environment attributes from local policy file, and the context handler gives the required target resource value, attribute and resource value to PDP (policy decision point) agent. Finally, the PDP decides access possibility and send context handler so that PEP agent allow or deny the request [19].

5.2 Flow of Certificate Validation Service in XKMS

Validation processing methods are composed of two steps. First, determination means accessing a repository retrieve the certificate and the construction of the path. Second, Validation means making sure that each certificate in the path has its integrity, is within its validity period; and has not been revoked. In CVM the client delegates subtasks (e.g. only path discovery) or the entire task (e.g. path discovery and path validation) of certificate path proce-ssing to a server, as it is depicted in fig. 8.

Path construction may require a path discovery task resulting in several certification paths found by the XKMS for a certain validate a certificate [23]. Execution of a path validation algorithm that includes certificate verification (i.e. whether it has expired or is revoked) for each certificate in the path and the processing of path constraints The algorithm must verify also the XML signature on each certificate, check that the required certificate policies are indicated in the certificates and check that the names in the certificates are consistent with a valid certification path, that is, the subject of every certificate in the path is the issuer of the next certificate (except the root CA).

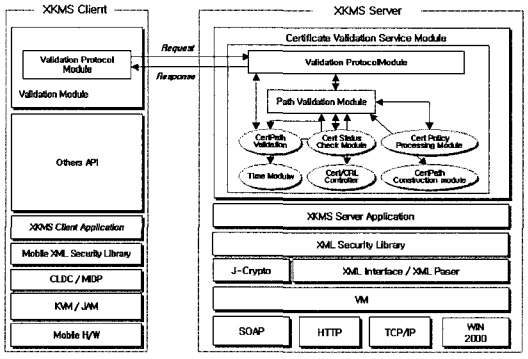

Fig. 8. CVM Components in XKMS.

As mentioned earlier, usually an OCSP responder signs online each response it produces. Responses can contain three times in them. 'ThisUpdate' is the time at which the status being indicated is known to be correct. 'NextUpdate' is the time at or before which newer information will be available about the status of the certificate. 'ProducedAt' is the time at which the OCSP responder signed this response.

The mechanism that we propose is called XKMS-OCSP and it exploits the fact that a one way hash function is at least 10,000 times faster to compute than a digital signature [24]. When a pre-produced response needs to be updated because its nextUpdate has become obsolete, a one way hash function is performed to update this response instead of a new signature. Using a one way hash function will permit the repository to update the responses more frequently without falling into denial of service. XKMS-

OCSP is based on the even et al. algorithm and it works as follows. When a response is going to be pre-produced, the responder adds a hash-chain to it. The hash chain permits the repository to update the pre-produced response in successive periods with scarce resources utilization. The hash chain results from applying d -- 1 times a one way hash function h over a secret nonce (1)

$$R \xrightarrow{h} R_d \xrightarrow{h} R_{d-1} \xrightarrow{h} \ldots R_i \xrightarrow{h} \ldots R_1 \xrightarrow{h} \ldots R_1 \xrightarrow{h} R_0 \quad (1)$$

Let us define the parameters involved in the process. PrimaryUpdateValue (R) is the secret nonce. R is only known by the responder (broker) and it is generated for each new pre-produced response. MaximumUpdateIndex (d) is the maximum number of periods that a preproduced response can be updated. BaseUpdateValue (R_0) is the last value of the hash chain and it is included in the signature computation of the pre-produced response. R_0 is computed by applying (d + 1) times h over R.

$$R_0 = h^{d+1}(R) \quad (2)$$

CurrentUpdate value (R_i) is computed by applying (d+1− i) times h over R

$$R_i = h^{d+1-i}(R) \quad (3)$$

Where i is the number of periods "Δ" elapsed from the documented one (the documented validity period is the period included in the response). Δ is defined as

$$\Delta = \text{nextUpdate} - \text{thisUpdate} \quad (4)$$

A relying party can verify the validity of a pre-produced response that it is living beyond its documented life-time, say, at time t, where t is included within the period [nextUpdate + (i − 1)Δ, nextUpdate + iΔ], by checking the equality of equation (5)

$$R_0 = h^i(R_i) \text{ with } i \le d \quad (5)$$

It must be stressed that to forge a currentUpdate value with the information provided by a previous update value an attacker needs to find a pre-image of a one way hash function which is by definition computationally infeasible.

Certification path validation verifies the binding among the subject identity, the subject public key, and subject attributes that may be present in the certification path [4,13,14]. Constraints in certification path limit the possible identity values and the possible attribute values. And certification path validation determines whether certificates in chain are revoked or not revoked. The algorithm of the validation is as follows.

(1) The client generates the certification path validation request. The request may optionally include the client's trust anchor or certification policy. This information is used for validating chain.
(2) The server builds certification paths using certification path construction module. If the trust anchor in the request is present, the server must build the certification path that start from the trust anchor certificate.
(3) Certification paths and optional certification policy.
(4) If building certification path succeeded in step 2 and certification policy is present, the server verifies certification path using certification policy constrains and also performs checking certificate status.
(5) If building certification path succeeded in step 2 and certification policy is not present, the validating process is performed by only checking certificate status. Because the current certification path was already verified in step 2.
(6) If the previous step succeeded and the verified path was obtained, the verified path is saved in certification path DB table for next request.
(7) If the previous step failed or the verified path was not obtained, the server processes an exception handling and makes a fail response. Also the fail reason is recorded in the log file.
(8) Verified certification path.

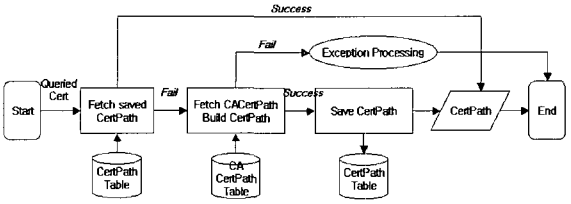

Fig. 9. Certification Path Validation Sequence.

6. Experiment of LBS Key Management Using XKMS

XKMS has been implemented based on the design described in previous section. Package library architecture of XKMS based on CAPI (Cryptographic Application Programming Interface) is illustrated in fig. 10. Components of the XKMS are XML security library, service components API, application program. Although XKMS service component is intended to support XML applications, it can also be used in order environments where the same management and deployment benefits are achievable. XKMS has been implemented in Java and it runs on JDK (Java Development Kit) ver. 1.3 or more.

The figure for representing Testbed architecture of XKMS service component is as follows Fig. 11. We use Testbed system of windows PC environment to simulate the processing of various service protocols. The protocols have been tested on pentium 3 and pentium 4 PCs. It has been tested on windows 2000 server, windows XP. The XKMS server is composed server service component of XKMS platform package. The communication protocol (e.g.

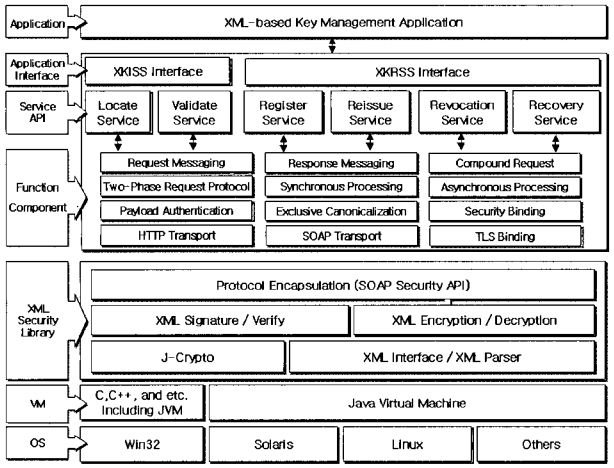

Fig. 10. Package Library Architecture of XKMS based on CAPI.

certificate validation protocol based on XKMS in open LBS) between the server and mobile client follows the standardized SOAP protocol illustrated in Fig. 12. And the message format is based on Specification of W3C (World Wide Web Consortium). Table 1 summarizes function of XKMS service system component for CVM.

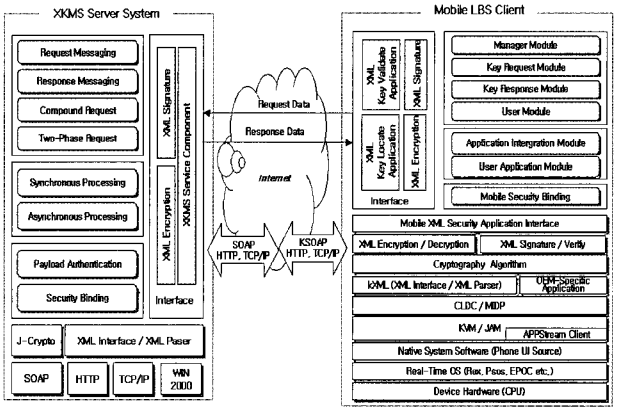

Fig. 11. Testbed Configuration of XKMS Component for Open LBS.

Fig. 12 showed difference for 0.2 seconds that compare average transfer time between client and server of XML encryption &decryption by XML signature base on XML security library. According as increase client number on the whole, showed phenomenon that increase until 0.3 seconds. Fig. 12 is change of average transmission time according as increase client number in whole protocol environment. If client number increases, we can see that average transfer time increases on the whole. And an average transfer time increase rapidly in case of client number is more than 45. Therefore, client number that can process stably in computer on testbed environment grasped about 40. When compare difference of signature time and protocol time, time of XML signature module is occupying and shows the importance of signature module about 60% of whole protocol time.

Table 1. Function of Key Management Component in Open LBS Service

Service & Protocol	XKMS system					
	Tier 0	Tier 1	Tier 2	Tier 3	Tier 4	
Register service	*	*	*	*	*	KRSS
Locate service	M	M	*	O	O	KISS
Validate service	M	M	M	O	O	KISS
Recovery / Revoke service	*	*	*	*	*	KRSS
Compound request protocol	O	O	O			-
Synchronous processing	*	M	M	*	*	-
Asynchronous processing	*	O	O	*	*	-
Two-Phase request protocol	*	O	O	*	*	-
Payload authentication	*	O	O	*	*	-
HTTP transport	M	M	M	M	M	-
SOAP 1.1 transport	M	M	M	M	M	-
Cert path validation	*	*	M	*	*	-
Cert status check	*	*	M	*	*	-

(M:Mandatory, O:Optional, *:No Recommendation)

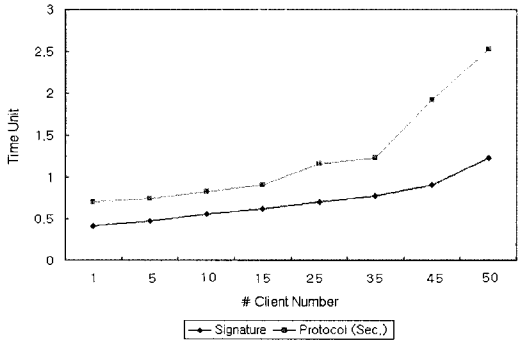

Fig. 12. Simulation Result of XKMS Server Protocol based on Open LBS.

7. Conclusion

Location-based services are so attractive that they can cover all walks of life. However, current LBS are growing slower than expected. Many problems like accuracy, privacy, security, customer requirement have to be addressed. It should be understood that there is no single universal solution to LBS.

We propose a novel security approach on open LBS to validate certificate based on current LBS security environment using XKMS and SAML, XACML in XML security. This service model allows a client to offload certificate handling to the server and enable to provide central administration of XKMS polices. In order to obtain timely certificate status information, the server uses several methods such as CRL, OCSP etc. Our approach will be a model for the future security system that offers security of open LBS security.

References

[1] XML Key Management Specification Version 2.0 (W3C Working Draft), April 2003.

[2] XML Signature Syntax and Processing (W3C/IETF Recommendation), February 2002.

[3] XML Encryption Syntax and Processing (W3C Recommendation), 2003.

[4] X.509 Certificate and CRL Profile, RFC2459, January 1999.

[5] M. Myers, R. Ankney, A. Malpani, S. Galperin, and C. Adams, X.509 Internet Public Key Infrastructure Online Certificate Status Protocol – OCSP, RFC 2560, June 1999.

[6] A. Malpani, P. Hoffman and R. Housley, Simple Certificate Validation Protocol, draft-ietf-pkix-scvp-09.txt, Jun 2000.

[7] D. Pinkas and R. Housley, Delegated Path Validation and Delegated Path Discovery Protocol Requirements, RFC 3379, 2002.

[8] Assertions and Protocol for the OASIS Security Assertion Mark up Language (SAML), OASIS Standard, 5 November 2002.

[9] A Selkirk, Using XML Security Mechanisms, BT Technol J 19 (3), 2001.

[10] Phillip Hallam-Baker, W3C XKMS Workshop position paper, Proceedings of XKMS Workshop, July 2001.

[11] E. Faldella and M.Prandini, A Novel Approach to On-Line Status Authentication of Public Key Certificates, in Proc. the 16th Annual Computer Security Applications Conference, Dec 2000.

[12] Y. Elley, A. Anderson, S. Hanna, S. Mullan, R. Perlman and S. Proctor, Building Certification Paths: Forward vs. Reverse, Proc. the Network and Distributed System Security Symposium Conference, 2001.

[13] M. Naor and K. Nissim, Certificate Revocation and Certificate Update, IEEE Journal on Selected Areas in Communications, 18 (4) (2000).

[14] Jonghyuk Roh, Seunghun Jin and Kyoonha Lee, Certificate Path Construction and Validation in CVS, KICS-Korea IT Forum, 2002.

[15] M. Prandini, Efficient Certificate Status Handling within PKIs: an Application to Public Administration Services, in Proc. the 15th Annual Computer Security Applications Conference, 1999.

[16] Donald E, Eastlake, Kitty Niles, Secure XML, Pearson addsion wesley, 2003.

[17] Blake Dournaee, XML Security, RSA Press, 2002.

[18] Haeock Choi, Open LBS Plarform Architecture, ISR2002, 2002.

[19] Euinam Huh, Jihye Kim, Hyeju Kim, Kiyoung Moon, Policy based on grid security infrastructure implementation for dirrerential resource access, ISOC 2003, 2003.

[20] Yuichi Nakamur, et. Al., Toward the Integration of web services security on enterprise environments, IEEE SAINT '02, 2002.

[21] OASIS, Web Service Security, http://www-106.ibm.com/, April 2002.

[22] Harsha Srivatsa, Location-based services, IBM Paper, November 2002.

[23] Diana Berbecaru, Antonio Lioy, Towards Simplifying PKI Implementation : Client-Server based Validation of Public Key Certificates, IEEE ISSPIT 2002, pp.277-281.

[24] Jose L. Munoz et. Al.,Using OCSP to Secure Certificate-Using transactions in M-Commerce. LNCS 2846 (2003) 280-292.

[25] Sungmin Lee et. Al., TY*SecureWS:An integrated Web Service Security Solution based on java, LNCS 2738 (2003) 186-195.

[26] Namje Park, Kiyoung Moon, Sungwon Sohn, XML Key Information System for Secure e-Trading, WSEAS TRANSACTIONS on COMPUTERS, 2 (2) (2003), 327-333.

Kiyoung Moon

He received his BS and MS degrees in electronics engineering in 1986 and 1989, respectively, from Kyungpook National University, Korea. He is PhD candidate in Computer Science at the Chungnam National University. He is a team leader of Biometrics Technology Research Team in Electronics and Telecommunications Research Institute (ETRI), Korea since 1994. His research interests include Web Services Security, Biometrics, and Bio-information Security.

Namje Park

He received the BS degree in information industry from Dongguk University, Korea in 2000, the MS degree in information security from Sungkyunkwan University, Seoul, Korea, in 2003. He is currently a member of the engineering staff in the Electronics and Telecommunications Research Institute (ETRI), Korea. His research interests are Mobile RFID Security, Grid Security, Key Management, and Location-based Service.

Kyoil Chung

He is a Director of Information Security Infrastructure Research Group at Electronics and Telecommunications Research Institute (ETRI), Korea. His main research areas are information security and IC Card&RFID Security. He received his B.E., M.E., and Ph.D. degrees in Information Engineering from Hanyang University in 1981, 1983, and 1997 respectively. His research interests include RFID Security, Biometrics, and Mobile Security.

Sungwon Sohn

He is currently the Executive Director of the Information Security Technology Division at Electronics and Telecommunications Research Institute (ETRI). He received his PhD degree in computer engineering from Choongpook University, Korea, in 1999. Since joining ETRI in 1991, his work has focused on the Network Security, Mobile Security, Active Network, and Biometry.

Jaecheol Ryou

He was born in Daejeon, Korea. He received the B.S. degree in industrial engineering from Hanyang University in 1985, the M.S. degree in computer science from lowa State University in 1988, and the Ph.D. degree in electrical engineering and computer science from Northwestern University in 1990. He joined the faculty of the Department of Computer Science at Chungnam National University, Daejeon, Korea, in 1991. His research interests include Computer and Communi-cation Security, and Distributed Processing.