PatientService : Electronic Patient Record Redaction and Delivery in Pervasive Environments

Amit Choudhri, Lalana Kagal, Anupam Joshi, Timothy Finin, Yelena Yesha
Computer Science and Electrical Engineering
University of Maryland Baltimore County
email : {amitc1, lkagal1, joshi, finin, yeyesha}@cs.umbc.edu

***Abstract*- E – healthcare today is moving away from the tethered domain and becoming diffused into an environment rich with portable digital devices. In this evolving environment, the need to deliver information such as Electronic Patient Records at the point – of – care is a prime factor in managing the healthcare system efficiently. This however presents serious security challenges in pervasive environments such as wirelessly – connected hospitals; where protecting the confidentiality of the information, while at the same time allowing authorized user to access it conveniently is the core issue in the paradigm. We describe the security challenges in pervasive computing environments, and explain why traditional security mechanisms fail to meet the demands of these environments. We use an architecture that incorporates policy based security and distributed trust management to provide a highly flexible approach for accessing Electronic Patient Records that are electronically redacted depending on the users digital credentials. We then present a prototype of the system using a variety of portable devices with wireless technology and include the policy used to test the system.**

I. INTRODUCTION

Healthcare issues are among the most critical ones facing our society today. There have been continual efforts over the years to apply all fields of science and technology to improve healthcare. The initiative all along has been to focus on providing healthcare than dealing with day-to-day tasks involved in doing so. As technology and communication shifts towards the mobile computing paradigm, a new field of healthcare called "Mobile Healthcare" or "M – Healthcare" has evolved.

M-Healthcare has been described as [1] "the application of mobile computing technologies to provide mobile access to healthcare information systems." M-Healthcare technologies enable access to important and useful information systems at the point of care, from remote locations, or from virtually any place within the healthcare facility. M-Healthcare covers a broad range of computing technologies that includes end user devices, network transport services, and application services as applied to environments such as hospitals.

Mobile computing allows us to extend the paradigm of availability and accessibility to computer resources without being tethered to a network. Though many enterprise applications exist for wireless applications, they rely on conventional security mechanisms such as authorization and static control lists to define the capabilities of different entities in the system, thereby limiting their extensibility. These methods use system-based controls to verify the identity of a person or process, explicitly enabling or restricting the ability to use, change or view a computer resource.

PatientService performs document redaction in a ubiquitous Healthcare environment. This service delivers different versions of documents to users based upon the their roles as defined by the *Privilege Management Toolkit (PMT)* [2] and any trust semantics that are in effect at that instant. We, therefore, focus on incorporating the human semantics of co-operative trust in an ad – hoc environment such as a hospital, so that the technology can match the intuitive security and access requirements in such environments and avoid the concept of static Access Control Lists, and instead base security on trust relationships and recommendations. Trust, in our system involves the delegation of permissions; and deciding a user's rights based on the policy and valid delegations in effect. This system does away with the manual blacking out of sensitive data in documents which may be viewed by people with varying security credentials.

Our system is capable of providing wireless connectivity with any of Infrared (IR), Bluetooth or 802.11b protocols. However, since Bluetooth and IR are highly restricted by their physical "line-of-sight" ranges and the number of devices that can be used simultaneously in the system, we prefer to use 802.11b as the network can exist in a larger environment such as a hospital using strategically placed base stations.

II. BACKGROUND, MOTIVATION & ISSUES INVOLVED

Hospital environments are typical of healthcare environments and encompass all the different scenarios that arise in healthcare from ER to out-patient care, billing, and the like. Up until now, e-health solutions have dealt with applications such as auto-prescription and remote clinical form filling in secure environments where ad-hoc networking is not a primary issue, and elementary end-to-end connectivity is all that is needed. In such environments, simple authentication and *Role Based Access Control (RBAC)* [3] schemes suffice to provide basic access control to resources in the system.

This research was supported in part by NSF Awards IIS 9875433 and CCR 0070802, the Defense Advanced Research Projects Agency under contract F30602-00-2-0 591 AO K528 and the IBM UPP award to Joshi.

However, patient medical records present a whole spectrum of sensitive issues relating to privacy and confidentiality. In the e-health paradigm, protecting Electronic Patient Records (EPR) [4] while allowing them to be accessed by administrators at all tiers in the environment is a primary concern. Modifications to the EPR by unauthorized personnel, or even basic read access to these without consent is an infringement of medical ethics, and in an ad-hoc pervasive environment like the one envisaged these problems are further compounded. This is because pervasive computing environments have several characteristics that existing security protocols are not suitable for; such as presence of foreign users, very large number of users and resources, difficulty in maintaining central control, and communication over insecure wireless links.

It is equally important that critical information regarding the patients such as their medical records and history be available to the physicians and nurses involved in their treatment, in a manner that is secure and also restricted on a need-to-know basis.

Currently, there are many vendors that provide database and file access schemes for EPR. However, these applications fail to work in a truly ubiquitous healthcare environment where users may need to delegate certain abilities as they do not incorporate the notion of co-operative trust. This is where *PatientService* makes an impact, and provides a shift within the M-Healthcare paradigm. Since we use a policy-based mechanism to specify our semantics, and use PMT permissions to represent roles in the system, we can dynamically modify the environment's representation in terms of the entities that exist in it, and their abilities as defined by their delegations and roles. This allows *PatientService* to be easily tailored to each healthcare environment.

Since the potential set of entity types in a healthcare environment such as a hospital is restricted to a few pre-defined types such as Doctors, Nurses, etc. it can be assumed that simply using Role Based Access Control could be a viable solution to providing access to different document sets for different roles. However RBAC alone cannot suffice as its principals are not suited to *Discretionary Access Control*. In RBAC all abilities that a role possesses are passed on to the entity it delegates rights to, since delegation is on a "per-role" basis, and one cannot merely delegate a subset of the abilities possessed by the delegator. Discretionary Access Control in this context is described [5] as "A means of restricting access to objects based on the identity of subjects and/or groups to which they belong. The controls are discretionary in the sense that a subject with a certain access permission is capable of passing that permission (perhaps indirectly) on to any other subject." This allows a smaller subset of abilities to be delegated to the other entity, thereby providing a stricter level of control.

We have tried to create an application that works on the following vision :
A doctor walking through a certain ward in a hospital approaches a patient's bed. The bed has a Bluetooth enabled sensor which recognizes the doctors PDA through a Bluetooth-based connection, and then retrieves that patients EPR on behalf of the doctor once the doctor's PDA sends his/her digital credentials to the sensor. Alternatively, existing geo-location based wireless technology can provide specific portions of the EPR for the patient that the doctor is closest to based on the doctors location in the hospital. e.g. information delivered to the doctor may differ while visiting a patient in the cardiac care unit, as opposed to visiting the same patient in a general recovery ward or an x-ray room.

Though these scenarios may seem complicated and futuristic, it is possible to extend our system to provide such functionality, thereby tremendously improving point-of-care information delivery.

III. Related work

There are a number of commercial projects that are aimed at providing E-healthcare solutions, especially those that deal with EPR's. However, none of them use distributed trust as a way to resolve complex security issues. Accordingly, after discussing some of the enterprise solutions to EPR, we will discuss some work carried out on distributed trust.

A. Electronic Patient Records – Related Work

Most commercial applications for EPR have been built upon the concept of using PDA's as micro-computers running enterprise software independently, and connecting to the central database over a wireless LAN connection. *Patient Tracker* [6] by HandHeldMed is a widely deployed patient charting application which allows mobile access to patient records and demographics among other reports. It uses simple password protection to protect patient records, and allows simple IR-based peer-to-peer transmission between *Patient Tracker* users. This ability to transmit records without being accountable to a central administration controlled security policy is an inherent flaw which can compromise the confidentiality of patient records.

Wireless MediCenter [7] provides a highly efficient solution tailored for portable devices such as PDA's and tablet PC's. It is described by its creators as being "a wireless, paperless electronic medical record (EMR) system." It uses read-write protection for access to the database where the EPR's are stored, and can deliver them over a secure LAN or through high-speed wireless connections, as used by portable devices. It is a comprehensive solution to all doctor and patient needs and provides different portals for the patients to review their information. However, it does not use any notions of trust or digital credentials such as digital certificates which can make

the entire application more secure, and also ease the workflow in the environment.

The *m-care* [8] project aims at providing secure access to patient records and other data using a WAP based architecture in conjunction with a WAP-based mobile phone. It uses Wireless Transaction Layer Security (WTLS) to provide network security and personal PIN numbers to provide access to the system. A Microsoft SQL Server holds static information about users and their access rights to the EPR database, and a simple firewall software is used on the server to restrict connections to the service from the WAP gateway. The use of PIN codes and static access control lists is not sufficient to deal with the accesses needed in a complex healthcare environment such as a hospital, though it may suffice for single individuals out in the field who can only access WAP services and cannot connect to the central database using a more secure wireless technology.

Other custom solutions such as *PatientKeeper* [9] and *PocketMD* [10] attempt similar methods to the above in order to provide EPR's on-the-go, but have identical shortcomings to the other applications discussed here.

Our system attempts to provide the semantics of human trust while still maintaining confidentiality of the records, and as such can be used as a module in any of the above applications to make them more secure and aligned with the intuitive trust that may exist in the healthcare environments that they are used in.

B. Distributed Trust – Related Work

Conventional authorization and access control schemes are no longer viable in mobile environments, as they suffer from communication overhead, implicit trust assumptions and access to a central location which is not always feasible or desirable.

Role Based Access Control is probably one of the best known methods for access control, where entities are assigned roles, and there are rights associated with each role. Unfortunately, this is difficult for systems where it is not possible to assign roles to all users and foreign users are common.

The Simple Public Key Infrastructure (SPKI) [11] was the first proposed standard for distributed trust management. This solution, though simple and elegant, includes only a rudimentary notion of delegation, which is crucial to the developed of *distributed trust*.

Blaze et al.[12],[13],[14] recommend a trust management approach, which formulates authorization as verifying whether an entity has the credentials that comply with the security policy governing the requested resource. These credentials include properties, and trust relationships of the entity with the other entities in the system. In this proposal, distributed trust is used as a way of authorizing users' requests based on a security policy which a Distributed Trust Manager reasons over to infer the trust relationships.

Using trust involves trust establishment, and trust management. Trust establishment is the process of deciding what the trust relationship with another entity should be. Trust management involves using the trust information, including recommendations from other trustees, to reason about authorization requests.

Blaze's PolicyMaker [12] is one of the first forays into distributed trust. PolicyMaker is able to interpret policies and answer questions about access rights. However, it is not easy to express policies in a simple manner in the same, which makes it difficult for non-technical administrators to use this tool effectively.

IV. ARCHITECTURE

The need to create a trust based model in a pervasive healthcare environment is the driving motivation behind the work carried out on *PatientService*. The trust management principles are enforced using the *Vigil* [15] infrastructure for ubiquitous environments. *Vigil* is an extension of the *Centaurus* [16] framework which provides portals to services using mobile devices. The *Privilege Management Tool (PMT)* authenticates the entities in the system and maps the digital credentials to a privileged role in the environment. Communication between the various modules of the system is carried out using messages written in *Centaurus Capability Markup Language (CCML)* [16] which is an application-specific extension of Extensible Markup Language (XML) [17].

The Vigil system is divided into *domains*, and each domain is controlled by one or more *Service Managers*. The *Service Manager* finds matching services for users. It allows users and services to register and then provides brokering between them. The *Communication Manager* provides a communication gateway between a Client and a Service Manager. Its sole purpose is to abstract and translate communications protocols. The *Certificate Authority* is responsible for generating x.509 version 3 digital certificates [18] for each entity in the system and for responding to certificate validation queries. The *Distributed Trust Manager* manages the trust in the domain, receiving information about new access rights that are conferred on a user, and reasoning about the current rights of a user. Finally there are users and services that are treated equally as *Clients*.

The clients initially register with a *Centaurus Service Manager* (SM) by sending a registration request message, and its credentials in the form of a x.509 digital certificate. The *Service Manager* in turn employs the *Certificate Authority* which verifies that the certificate is valid. The SM then allows

the client to connect to it and become a part of the pervasive environment by sending it a registration response.

Next, the *Service Manager* sends the client a list of all services available in the environment to which the client can subscribe depending on its credentials. Notable among these services is *PatientService*, to which all clients can subscribe, as long as they are connected to the Service Manager located in the environment . However, this merely allows the clients to be aware of and able to subscribe to the *PatientService*, not to obtain the EPR from it. In restricted healthcare environments such as military hospitals, clients without sufficient credentials may not even be informed of the existence of such a service.

Once registered and provided *PatientService* as an accessible service, a client can subscribe to *PatientService* by sending a subscription request for the same to the SM. The SM recognizes the request and forwards the request to the PMT module. The PMT module manages privileges for an incoming request and maps those privileges into permissions, which are akin to the clients roles in a RBAC system. The PMT returns this permission to the SM which then passes it on to the *Distributed Trust Manager* (DTM) .

The DTM is responsible for maintaining distributed trust in the Vigil system. It interprets the organization's security policy in order to provide controlled access to Services and uses distributed trust as a more flexible and easily extensible policy-based mechanism. A policy includes rules for role assignment, rules for access control, and rules for delegation. We define a policy after Bradshaw et. al [19] as "an enforceable well-specified constraint on the performance of a machine-monitored action by a subject in a given situation." This definition is further explained in Fig. 1. by describing its constituents.

- *Enforceable:* In principle, an action controlled by policy must be of the sort that it can be prevented, monitored, or enabled
- ***Well-specified:*** Policies are well-defined declarative descriptions
- *Constraint on the performance:* The objective of policy is to assure, with or without the knowledge or cooperation of the component being governed, that the policy administrator's intent is carried out with respect to whether the specified action takes place or how it takes place. Policy captures a set of general principles and constraints that can be applied to specific, even novel situations while assuring the policy makers intent
- ***Machine-monitored action:*** Generally the actions governed will be machine-executable ones, but we are also interested in dealing with situations where a person is responsible for completing an action and then somehow signaling that fact to the machine
- ***Subject:*** The subject is either a human or a hardware or software component-or some group of such entities
- ***Situation:*** Policy applicability may be determined by a variety of preconditions and contextual factors.

Figure 1. Bradshaw et. all's itemized constitutions of a policy

This policy can be extended by the use of *delegation* by authorized entities to other entities. These delegations are only valid if the delegator has the right to delegate. Revocation of rights is also possible, which allows for "restoring" the security semantics back to their default interpretation once the delegations have been revoked. The ability to modify such policies with ease allows for flexibility in re-defining the "trust semantics" of the current environment, which are highly dynamic. With *PatientService*, a user such as a visiting specialist without certain access rights may be granted those rights for a certain period of time by another user such as a resident doctor that is capable of making delegations, thereby overcoming the need for the resident doctors physical presence.

After the DTM has analyzed the clients permissions, and verified its capabilities depending upon the current security policy and delegations in effect, it obtains a "clearance level" for the client from the policy interpretation. This clearance level is then applied as a parameter to the XSLT [20] filtering module. Thus, our system uses a 3-tier verification and authorization mechanism to enable the client to obtain a record. All of this is carried out without the user's knowledge who merely submits a registration request to the SM with his/her digital certificate, and attempts to subscribe to the *PatientService*.

The XSLT filtering module is responsible for applying the appropriate XML translations to the required document, corresponding to the clearance level passed in to it by the DTM. The document root for the patient's record is pre-tagged in XML. The filter module parses this document, editing out any content marked with a tag associated with a clearance level higher than that passed in by the DTM. The mechanism for this functionality is illustrated in Fig. 2.

This "redaction" procedure is simple, but efficient. The documents can be tagged at varying levels of granularity, from the entire document base itself having a single tag, to phrases and individual fields being tagged separately. This allows for accentuated control in environments that have stricter interpretations for access to various fields of an EPR. Also, to decrease the turn-around time in delivering the redacted version of the document, we can have the filter "pre-parse" the documents in the system repository and maintain document trees for each patients documents. The leaves of the tree would then correspond to the different documents obtained by parsing the original root document for all possible clearance levels.

Now, when the DTM passes in a given clearance level to the filter module, the required document can be returned immediately by simply tracing the appropriate path down the XSLT tree for the given clearance level, and returning the leaf node for that path. Although this sounds like a practical approach to reduce latency in the system, it requires storing multiple copies of a single document, and the storage required for a large number of patient records may not scale favorably.

Thus, the classical latency v/s. storage space trade-off exists in our application as well, and the option to pre-parse or not is an environment-specific choice to be made by the healthcare administrators in that environment.

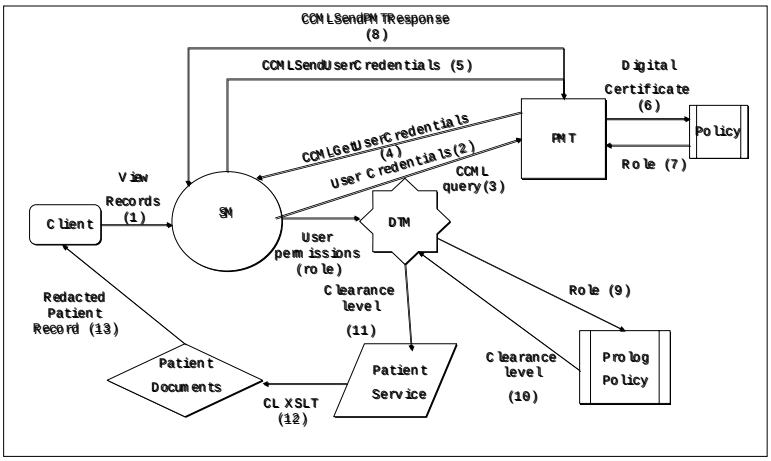

Figure 2. PatientService mechanism for document redaction

V. IMPLEMENTATION

While our system deals with security at a higher level of abstraction, we rely on standard encryption procedures such as WEP [21] to ensure that confidential data is not intercepted and decrypted during physical transmission. Consequently, we did not investigate lower level security issues in our implementation and experiments.

In order to test the use of *PatientService* in a mock healthcare environment, we created a preliminary scenario with different roles and varying levels of clearance, along with the rules for delegation and trust as defined in the Prolog policy in Fig. 3. This is the policy that the DTM interprets to allocate a clearance level for the documents.

In the policy shown in Fig. 3, all roles are associated directly or indirectly to a patient, Joe, whose records are to be viewed using *PatientService*. The roles have their own static clearance levels between cl2 and cl4 depending upon the need-to-know information about from the EPR. A delegatedToUser statement allows a role to delegate a certain clearance level to another role in the system provided the required permittedToDelegate, associatedWith and tendedBy relationships hold. e.g. a Doctor can delegate a clearance level of cl4 to a visiting Specialist for a particular patient's (Joe's) EPR, provided the Doctor is permitted to delegate the clearance level, and is the doctor tending to that patient.

The document root for the patient records is comprised of all the details that could constitute the patients personal information, medical history, insurance information and current medical status. The records to be accessed are available to a central server, and the pages have been pre-edited depending upon the clearance level to reduce turn-around time. i.e. four records of each patient record are available at the server, each corresponding to one of the four clearance levels

```
:- dynamic(agent/1).
:- dynamic(role/2).
:- dynamic(permittedToDelegate/3).
:- dynamic(delegatedBy/3).
:- dynamic(delegatedToUser/7).

agent('doctor')
role('Doctor').
permittedToDelegate('Doctor','Nurse',cl4).
permittedToDelegate('Doctor','Specialist',cl4).

agent('nurse')
role('Nurse').
associatedWith('Nurse','Doctor').

agent('specialist').
role('specialist','Specialist').

agent('hs').
role('hs','Hospital Staff').

agent('visitor').
role('visitor','Visitor').

agent('Joe').
role('Joe','Patient').
tendedby('Joe','Doctor').

clearance(HStaff,Patient,cl4) :-
        role(HStaff,'Doctor'),
        role(Patient,'Patient'),
        tendedby(Patient,HStaff).


clearance(HStaff,Patient,cl4) :-
        role(HStaff,'Nurse'),
        role(Patient,'Patient'),
        delegatedToUser(Doctor,DLoc,HStaff,ALoc,Patient,_,Time),
      permittedToDelegate(Doctor,'Nurse',cl4),
        role(Doctor, 'Doctor'),
        associatedWith(HStaff,Doctor), tendedby(Patient,Doctor).

clearance(HStaff,Patient,cl4) :-
        role(HStaff,'Specialist'),
        role(Patient,'Patient'),
        delegatedToUser(Doctor,DLoc,HStaff,ALoc,Patient,_,Time),
        role(Doctor, 'Doctor'),
      permittedToDelegate(Doctor,'Specialist',cl4),
        tendedby(Patient,Doctor).

clearance(HStaff,Patient,cl3) :-
        role(HStaff,'Nurse'),
        role(Patient,'Patient'),
        tendedby(Patient,Doctor),
        role(Doctor, 'Doctor'),
        associatedWith(HStaff,Doctor).

clearance(HStaff,Patient,cl2) :-
        role(HStaff,'Hospital Staff'),
        role(Patient,'Patient').

clearance(HStaff,Patient,cl1) :-
        role(HStaff,'Visitor'),
        role(Patient,'Patient').
```

Figure 3. Sample Prolog policy used in implementation.

and containing different versions of the content for the same page, and available as leaves of an XSLT tree. The electronically redacted document is then returned to the users PDA or Tablet PC.

VI. EXPERIMENTS

In order to test *PatientService*, we used portable devices such as two Compaq Ipaq's and a Tablet PC's interacting in a 802.11b-based wireless environment, each provided with a digital certificate corresponding to a different role, viz. Doctor, Nurse and Specialist. After registering and being authorized to access *PatientService,* all 3 roles requested access to patient Joe's EPR, and received different pre-parsed versions of the same corresponding to clearance levels cl4, cl3 and cl1, with cl4 being the highest of the three and showing the most information, and the Specialist being a Visitor simply received basic cl1 access.

We then delegated clearance level cl4 from the Doctor to the Nurse, as well as cl4 from the Doctor to the Specialist. This enabled both the Nurse and the Specialist to access Joe's records at cl4 for the time duration specified by the Doctor during delegation, after which these elevated clearance levels were revoked and the default policy implementation went back into place, returning the Nurse back to cl3 and the Specialist back to cl1.

VII. CONCLUSION AND FURTHER WORK

When a pervasive environment exists in a complex healthcare organization such as a hospital, the security needs for protecting sensitive data such as Electronic Patient Records becomes a key concern in the design of the environment. The presence of foreign users, very large number of users and resources and restricted ability to apply central control in the system makes much of the existing research in the area of security of distributed systems inadequate for pervasive environments.

PatientService uses a distributed trust approach to provide security in pervasive computing environments by combining elements of authentication via digital certificates, permissions obtained using the Privilege Management Toolkit and security policies that are reasoned over to create a multi-tiered access control mechanism that incorporates the trust semantics of delegation and revocation. This allows for greater flexibility in access control over services in pervasive environments.

Our infrastructure allows organizations to develop security policies that are flexible, easy to control and easy to implement while still strictly providing adequate access control to services. This eases the task of healthcare administrators, while still maintaining the confidentiality of documents.

To provide a broader application based on our concept of *PatientService,* we also plan on developing a web server based application using PMT. This allows global access as it is not restricted by the technologies used by the Centaurus/Vigil system, and will provide a convenient means of *document redaction* in healthcare and military environments using the web. The proposed concept is to incorporate *PatientService* into an adaptive web browsing scheme. In such an application, the content being displayed for a given URL will differ from user to user depending on the role(s) that the user can be accredited with on the basis of his/her digital certificate and the site's policy. The policy interpretation in this case is carried out by the PMT module instead of the DTM. Here, the PMT reads a policy file akin to that by the DTM, and can assign *permissions* in addition to the roles, which are used for trust delegations and revocation semantics.

In order to access an EPR, the user would present a digital certificate along with the URL for the required web page to a web server. The server would then pass on the certificate and URL to a "filter" module. This module would invoke the Policy Management scheme present in PMT to authenticate the user, and classify the user into the appropriate clearance category, using the roles and permissions inferred by PMT. This mechanism is shown in Fig. 4.

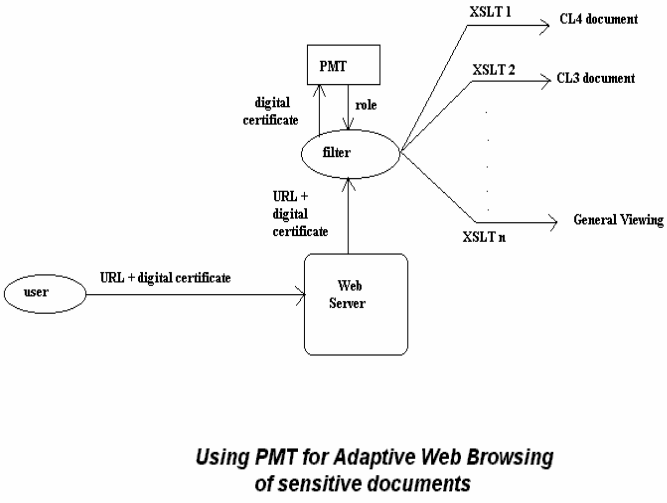

Figure 4. PatientService used as an adaptive web browsing mechanism for documents.

Once in the wired web domain, we can further extend this concept using the principles of the "Semantic Web", wherein the web pages to be viewed have some semantic markup such as DAML [22] or RDF [23] instead of just simple XML. Now, instead of "pre computing" the content for a web page, the content can be automatically generated "on the fly ". In this case, the filter module calculates the clearance and access right, and parses the page for the semantic information. It then displays only that content whose markup allows them to be displayed for the requesting users role, and asserts facts about the users when the document is parsed for its semantic meaning.

Since *PatientService* uses a policy-driven trust mechanism to determine permissions and clearance levels, we believe that it

can be easily adapted for document redaction needs in contexts other than just healthcare. A simple vision is that of a military application wherein documents are classified using tags corresponding to General Viewing, Confidential, Secret and Top Secret levels of security. Although, here we use simple clearance model for access classification, it is not too difficult to extend it to function to more complex access models such as lattices. Here, the redaction process is simplified in that PMT simply maps the already available security clearances assigned to the military personnel to the corresponding clearance levels in the policy. The DTM then incorporates any trust-based delegations and revocations active at that instant to return the appropriately redacted document. This adaptability across a multitude of contexts using the pervasive computing paradigm makes the architecture behind *PatientService* a highly viable solution to document redaction and delivery needs.

PatientService uses well-defined Prolog policies to specify the security constraints for the system. If the HIPAA requirements were to be made available in an arbitrarily complex Prolog policy, *PatientService* would then enforce those requirements, thereby making our system HIPAA-compliant as well. We see this conformance with federal and other organization-specific standards as being critical in the e-healthcare industry and our systems ability to adhere to these requirements also makes it a commercially viable option.

ACKNOWLEDGMENT

The authors of this paper would like to thank Dr. Tom Karygiannis, Dr. Wayne Jansen, Serban Gavrila, Vladimir Korolev and Thomas Heute at the National Institute of Standards and Technology (NIST) for use of their Privilege Management Toolkit (PMT) and invaluable assistance on this project.

REFERENCES

[1] Ardea Technology Group Inc. White paper on "The Emergence of M – Healthcare".
[2] W. Jansen, National Institute of Standards and Technology. A Privilege Management Scheme for Mobile Agent Systems. In the First International Workshop on Security of Mobile Multiagent Systems, Autonomous Agents Conference, May 2001.
[3] David Ferraiolo and Richard Kuhn. Role based access control. In *Proceedings of the 15th National Computer Security Conference*,1992.
[4] The Joint Computing Group of the General Practitioners' Committee and the Royal College of General Practitioners. Good Practice Guidelines for General Practice: Electronic Patient Records.
[5] National Computer Security Center. A Guide to Understanding Discretionary Access Control in Trusted Systems. http://comsec.theclerk.com/CISSP/neon_orange.htm .
[6] HandHeldMed. Patient Tracker Products. http://www.patienttracker.com/products.htm
[7] Wireless Medicenter. http://www.wirelessmedicenter.com/mc/glance.cfm
[8] David Brazier, Alpha Bravo Charlie Ltd. The m-care project. http://www.m-care.co.uk/tech.html
[9] PatientKeeper . http://www.patientkeeper.com/products.html
[10] PocketMD . http://www.pocketmd.com/white_papers_text.asp?wpID=6
[11] IETF. Simple public key infrastructure (spki) charter: http://www.ietf.org/html.charters/spkicharter.html.
[12] M.Blaze, J.Feigenbaum, and J.Lacy. Decentralized trust management. *IEEE Proceedings of the 17th Symposium*, 1996.
[13] M. Blaze, J. Feigenbaum, J. Ioannidis, and A. Keromytis. The role of trust management in distributed systems. Secure Internet Programming, LNCS vol. 1603, Springer, Berlin, 1999, pages 185-210, 1999.
[14] M. Blaze, J. Feigenbaum, J. Ioannidis, and A. Keromytis. The keynote trust management system version. Internet RFC 2704, September 1999, 1999.
[15] Lalana Kagal, Jeffrey Undercoffer, Filip Perich, Anupam Joshi, Tim Finin and Yelena Yesha. Vigil : A Secure Infrastructure for Service Discovery and Management in Pervasive Computing. In ACM MONET: "The Journal of Special Issues on Mobility of Systems, Users, Data and Computing ", April 2003.
[16] Lalana Kagal, Vladimir Korolev, Sasikanth Avancha, Anupam Joshi, Timothy Finin, and Yelena Yesha. Centaurus : An Infrastructure for Service Management in Ubiquitous Computing. In ACM Wireless Networks Journal, Volume 8, November 2002..
[17] W3C. Extensible markup language.
[18] R. Housley, W. Ford, W. Polk, and D. Solo. RFC 2459 Internet X.509 Public Key Infrastructure Certificate and CRL Profile. January 1999.
[19] Jeff Bradshaw, Mark Hoffman, James Just, and Mike Bennett, DARPA Policy Management Workshop Orientation, unpublished, March 2003.
[20] W3C. XSL Transformations.
[21] Nikita Borisov, Ian Goldberg, and David Wagner. Security of the WEP Algorithm. http://www.isaac.cs.berkeley.edu/isaac/wep-faq.html
[22] The DARPA Agent Markup Language. http://www.daml.org/
[23] W3C. The Resource Description Framework.