# Towards Scalable Authentication in Health Services

Gail-Joon Ahn and Dongwan Shin
*College of Information Technology*
*University of North Carolina at Charlotte*
*{gahn,doshin}@uncc.edu*

## Abstract

*Over the last two decades, many attempts have been made to computerize the management of patient records using advanced computing and networking facilities across healthcare providers such as hospitals, clinics, and clearing agencies. In addition to this transition from disparate and paper-based infrastructure to consolidated and digital medium-based one, we have been confronted with privacy and security requirements since the advent of the Health Insurance Portability and Accountability Act (HIPPA). The problem we seek to address in this paper is to provide authentication of individual identity in the context of accessing critical information in web-based e-Health systems including secure transmission of data across the Internet. These problems have technical solutions that are well known, but the solutions in general are strongly biased toward a single individual interacting with a single application. In this paper, we propose a scalable token-based authentication architecture and demonstrate how we can implement this architecture using commercial-off-the-self technologies. Our approach focuses on vendor-neutral specifications. The proof-of-concept prototype has been implemented so that the pilot testing may be conducted at various sites.*

## 1. Introduction

The healthcare industries have recently faced with the challenge of reducing costs by removing paper and manual processes to electronic and automated processes. Also, today's healthcare industry is striving to achieve the development and deployment of computer-based patient records for improvements in healthcare quality, cost, and access. In order to achieve this objective, vendors and healthcare organizations have been implementing various comprehensive healthcare systems.

A computer-based patient record is electronically maintained and it includes information about an individual's lifetime health status and healthcare. It facilitates for legitimate users to access health data stored in multiple, dispersed locations. Such access to data contributes to continuity of healthcare. Integration of health data with administrative and legal data supports the business operations of the healthcare provider and secondary uses such as insurance authorization, claims adjudication, and reimbursement; public health reporting; and continuing education. Enhanced communication among caregivers and documentation of an individual's health status also support consumer education and continuity of care.

While computerized patient record management has achieved modest progress--facilitating seamless integration with organization-wide health information systems, its provision for security and privacy has been extremely demanded.

In e-Health systems, the following security requirements have been widely addressed [1,14]:

- An individual should be strongly authenticated and authorized to access their protected health information.
- An individual should have a right to inspect and obtain a copy of their protected health information, or should have a right to control access to their protected health information.
- Information stored or in transit should be protected from various attacks such as virus and wiretapping.

In this paper, we focus on the first requirement, especially on a*uthentication* that is one of fundamental security services. Authentication is the process to verify the identity of an entity (i.e., a user) that wants to access another entity (i.e., a system). Traditionally, most distributed computing systems have relied on password-based authentication, which can be easily cracked. According to National Institute of Standards and Technology (NIST) study, a strong authentication should be characterized by the use of at least two kinds (or pieces) of evidence [3]. For example, a user might be authenticated by his/her

Proceedings of the Eleventh IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE'02)
1080-1383/02 $17.00 © 2002 **IEEE**

password combining with a token-based evidence, or by a token-based evidence with his/her biometric information. This immediately motivated us to use smart tokens (i.e., tokens with computing capabilities to support cryptographic functions such as encryption, digital signature, and key agreement) for our authentication framework.

In this paper, we propose a scalable token-based authentication framework for e-Health systems and attempt to deal with following objectives:

- Token-based authentication procedures should be relatively easy for users to utilize.
- A user (i.e., patient or doctor) typically needs a set of services from different service providers, thus his/her token should be able to communicate with different authentication services from different service providers.
- A service provider (i.e., hospital, insurance company, or billing agency) may need to support a set of different smart tokens that might be used by users.

The rest of this paper is organized as follows. Section 2 outlines background technologies. Section 3 introduces the components of our scalable token-based authentication framework. In Section 4, we demonstrate the feasibility of our approach through the proof-of-concept implementation. Section 5 concludes the paper.

## 2. Background Technologies

### 2.1 Authentication Protocols

Authentication protocol is a sequence of message exchanges between principals [5]. Various authentication protocols for distributed network systems have been proposed and implemented [6,7,15]. In particular, [6] surveyed the various forms of authentication protocols based on cryptographic technologies.

### 2.2 Public Key Infrastructure (PKI) and Digital Signature

PKI is an infrastructure for disseminating the public key in a secure and reliable channel. And it includes a set of components that manages certificates and keys used by encryption and digital signature. Public key encryption is the process that a sender transforms plaintext data into unintelligible data with a receiver's public key. The receiver can decrypt the message with his/her private key. Digital signature is the process that a sender signs the message with his/her private key. The receiver can verify the signature with the sender's public key. One of important components of PKI is a set of certificate authorities (CAs) that archives public keys of certified users or entities. The user or entity that wishes to participate in this infrastructure must successfully prove their identity to the CA.

Even though some argued the risks on security services of PKI [2], PKI has been considered as a viable solution for security and privacy services by healthcare industries. Hence, our work utilizes PKI to develop a scalable token-based authentication framework for healthcare systems.

### 2.3 Smart Tokens

Smart tokens are devices with a memory and a processor which can generate and store keys. It also supports cryptographic functions such as encryption, digital signature, or key agreement. Some noticeable characteristics of smart tokens are portability, tamper-resistant storage, and isolation of computational activities (i.e. leveraging the features of cryptographic functions without revealing private keys to other system components). [4]

### 2.4 Smart Token Technologies

**2.4.1 PKCS #11.** PKCS #11 (or Cryptoki) is an application programming interface designed and specified by RSA Laboratories to solve incompatibility problem between cryptographic applications developed by different vendors [11]. The primary goal of Cryptoki is to abstract the details of the devices through a lower level programming interface so that each application can have a common model to access cryptographic tokens. A secondary goal is to provide resource sharing. That is, a single device can be shared by more than one application.

**2.4.2 Java Card.** Java Card technology is developed by Sun Microsystems Inc. And the Java Card platform provides portability across various Java implementations. This "write once, run anywhere" capability is the most significant feature of the Java Card framework.

Java Card technology enables Java programs to be run on smart cards and other small, resource-constrained devices. Developers can build and test programs using standard software development tools then they are converted into a form that can be installed into a Java Card technology enabled device. Application software for the Java Card platform is called as Java Card applet or card applet (to distinguish it from browser applets).

### 2.5 Secure Socket Layer (SSL)

SSL [12] was introduced with the Netscape Navigator browser in 1994, and rapidly became the predominant security protocol on the Web. Since the protocol operates at the transport layer, any program that uses TCP (Transmission Control Protocol) is ready to use SSL connections. The SSL protocol provides a secure means for establishing an encrypted communication between Web servers and browsers. SSL also supports the authentication service between Web servers and browsers. SSL uses X.509 certificates. Server certificates provide a way for users to authenticate the identity of a Web server. The Web browser uses the server's public key to negotiate a secure TCP connection with the Web server. Optionally, the Web server can authenticate users by verifying the contents of the client certificates.

### 2.6 Java Plug-in

Java Plug-in enables users to direct applets or JavaBeans on their web pages to trigger Sun's Java 2 Runtime Environment (JRE), instead of the web browser's default Java virtual machine. Typically Java Plug-in programs are digitally signed by developers so that the users on the client side can check the integrity of Java Plug-in programs through signature verification and establish the trustworthiness.

## 3. Scalable Authentication Framework

Figure 1 illustrates our scalable authentication framework and the relationship between its core components: *token interface API*, *server interface API*, and an *intermediary authentication module*.

**Token Interface API** defines the communication interface to establish a channel between different types of smart tokens and intermediary authentication module. This provides a unified API based on the underlying implementation of smart tokens provided by manufacturers or vendors.

**Intermediary Authentication Module** (IAM) is a component running on the remote host to which smart tokens' interfaces are attached. This module handles a user's interaction with smart tokens and the user's login process.

**Server Interface API** defines the communication between different types of authentication services and IAM. Authentication services may include various forms such as signature-based, password-based, or biometrics-based authentication.

In summary, IAM mediates authentication processes to support various smart tokens and various authentication methods through token/server interface APIs.

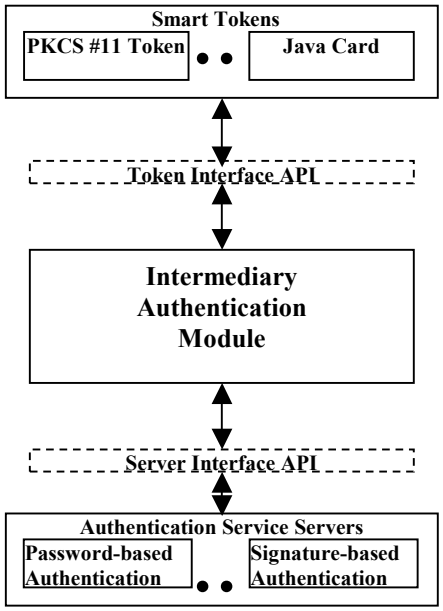


**Figure 1. Components of Framework**

### 3.1 Architectural Components

Based on the above-mentioned framework, we introduce a system architecture as shown in Figure 2. It includes architectural components and their interactions. For simplicity, we categorize these components into two groups: client-side components and server-side components.

**Client-side components**:

- Client: Client could be a web browser or other client applications and it downloads the intermediary authentication module. It should have a capability to establish a secure channel with server. And the client allows an intermediary authentication module from server to access system resources such as token drivers or system libraries.

- Smart Token/Reader: Passwords, private keys and corresponding public key certificates, or other credentials can be stored in smart tokens. Token reader could be a vendor specific reader such as smart card, serial port, or USB port readers.

- Intermediary Authentication Module (IAM): This consists of mobile executable codes, which could be easily loaded into the remote client from the server.

**Server-side components**:

- Server: Server could be a web server or other authentication server that is able to communicate with IAM. Like the client, the server must be armored with a capability to establish a secure communication channel with the client.

- Time server: Time server is utilized to help server generate a timestamp-based challenge for authentication procedures. Time server sends the current GMT time to the server when it receives the request from the server.

- Directory service/CA: Directory service is to store users' public key certificates for offline verification. Certification Authority is for issuance/revocation of certificates and online verification.

### 3.2 Authentication Procedure

Our authentication procedure first initiates SSL protocol to establish the secure communication channel between a client and a server. Through SSL, the client authenticates the server. Our authentication process continues only if the client successfully authenticates the server.

After the successful SSL authentication, the signed IAM is downloaded from the server. It could be ActiveX technology or Java Plug-in technology. Upon the verification of signature, the IAM can communicate with the smart token. The IAM loads necessary libraries and token communication packages, which are compliant to the token interface API in our framework.

Next, the smart token gets the server's challenge that consists of a random number and a timestamp. Then it signs the challenge with the user's private key stored in the smart token. The signed challenge is sent to the server as a response. The server verifies digital signature on the response to authenticate the user.

## 4. Implementation Details

To demonstrate the feasibility of our proposed scalable authentication framework, we implemented the authentication service with existing COTS technologies. We used Microsoft Internet explorer 5.5 and Netscape Navigator 4.7 for the client component. For the server component Apache Web server (1.3.24) with Java Servlet container (Apache Tomcat 4.0) was utilized. We selected Netscape Directory server 4.0 and Web server GMT timer for the directory server and the time server, respectively.

The intermediary authentication module was developed by using Java Plug-in technology. Figure 3 shows the interface of IAM. Sun's DSA signature mechanism provided in Java SDK 1.3 was used to sign IAM. The server API was implemented by Java Servlet technology to handle the communication with IAM. In addition, we used Cryptix RSA provider module on the server side.

Smart tokens used in our implementation are iButton and iKey, for Java Card framework enabled and PKCS #11 compatible tokens, respectively. We developed software adaptors for the smart tokens to work on our framework because the libraries from token vendors are not designed to work with an integrated framework [8,9,10]. These adaptors were written in Java. The hardware interface for iButton is a serial cable reader, whereas iKey uses USB port. Both tokens provide the basic RSA cryptographic functions such as encryption and digital signature. For reliability and compatibility purposes, we used Java version of PKCS #11 wrapper from IBM, which provides the interfaces for the native PKCS codes from the vendors.

For brevity, we assumed that the authentication server should be physically strongly protected by insider attacks or outsider attacks. Our prototype implementation has been recognized as a recommended solution by NC Department of Health and Human Services [13].

## 5. Discussion and Conclusion

In this paper we described a token-based scalable authentication service for e-Health systems. Our authentication service is based on the digital signature that public key cryptographic system provides and it supports multiple smart tokens. It also works transparently on between the remote server and tokens through an intermediary login component. We have developed the proof-of-concept implementation to demonstrate the feasibility of our approach.

With improved authentication mechanisms, we believe that institutions holding sensitive or protected critical information will increase the confidence of users, providers, and the public at large that the information is appropriately protected from unauthorized use or disclosure. Also, our novel

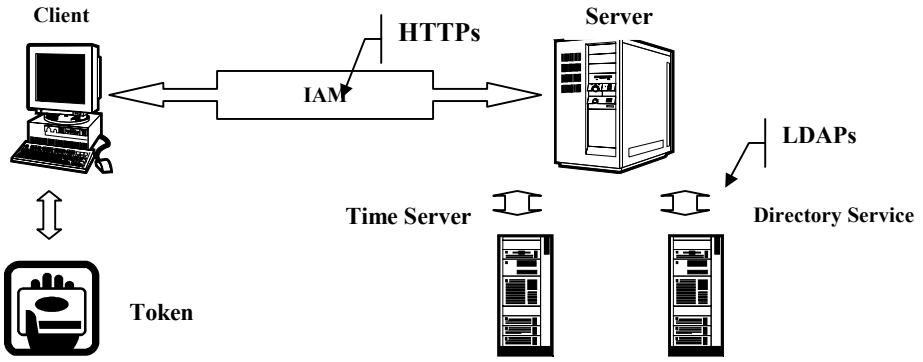


**Figure 2. Architectural components**

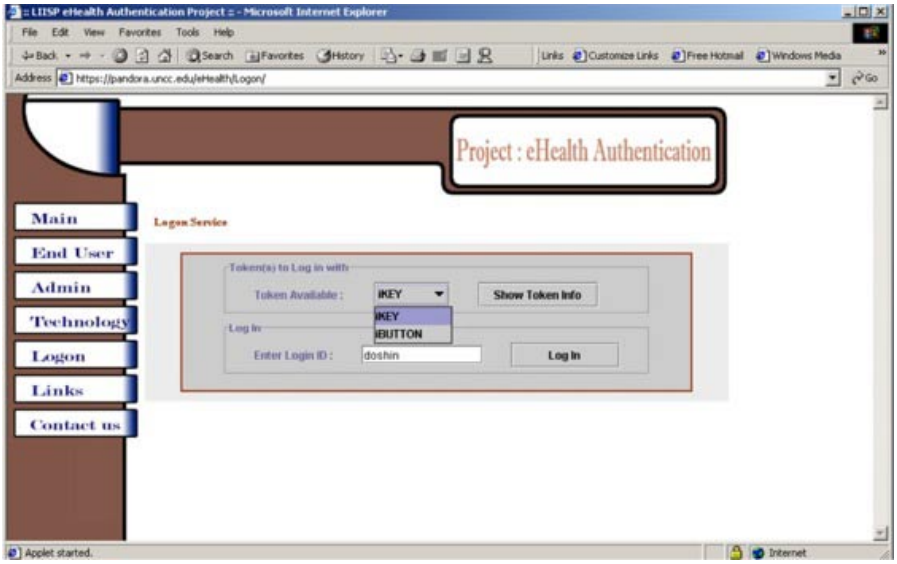


**Figure 3. Token-based authentication interface**

framework to provide a systematic approach will provide the key that enables new business models for critical information systems to safely, easily and quickly enter the Internet world.

Authentication is often combined with access control, which is the process to determine the authenticated user's privilege in the system or service. We are currently working on how we can leverage our framework for an attribute-based access control using X.509 attribute certificates stored in a smart token.

**Acknowledgement**

This work was partially supported at the Laboratory of Information of Integration, Security and Privacy at University of North Carolina at Charlotte by grants from the National Science Foundation (NSF-CCR-0124873) and the Robert Wood Johnson Foundation.

## 6. References

[1] Health Insurance Portability and Accountability Act of 1996. Available at http://frwebgate.access.gpo.gov/

[2] C. Ellison and B. Schneier. Ten Risks of PKI: What you are not being told about Public Key Infrastructure. *Computer Security Journal*, 16(1):1-7, 2000.

[3] NIST FIPS 190. Guideline for the Use of Advanced Authentication Technology Alternatives. September 1994

[3] H. Gobioff, S. Smith, J. D. Tygar, and B. Yee. Smart Cards in hostile environments. *In Proceedings of The Second USENIX Workshop on Electronic Commerce*, Oakland, CA, 1996.

[4] T. Stabell-Kula, R. Arild, and P. Harald. Providing Authentication to Messages Signed with a Smart Card in Hostile Environment. *USENIX Workshop on Smartcard Technology*, Chicago, USA, 1999.

[5] M. Burrows, M. Abadi, and R. Needham. A Logic of Authentication. *Technical Report* 39, Digital Systems Research Center, February 1989.

[6] John Clark and Jeremy Jacob. A Survey of Authentication Protocol Literature: Version 1.0. *York University Technical Report*, November 1997.

[7] M. S. Hwang and L. H. Li, "A new remote user authentication scheme using smart cards," IEEE Transactions on Consumer Electronics, 46(1):28-30, February 2000.

[8] PKCS #11 compatible smart card. URL: http://www.gemplus.com.

[9] PKCS #11 compatible iButton. URL: http://www.iButton.com

[10] PKCS #11 compatible iKey. URL: http://www.rainbow.com

[11] RSA Laboratories. PKCS #11 v2.11 Draft 1: Cryptographic Token Interface Standard. November 2000.

[12] Transport Layer Security Working Group. INTERNET DRAFT: The SSL Protocol Version 3.0. November 1996.

[13] North Carolina Department of Health and Human Services. Available at http://www.hipaagives.org

[14] For the Record: Protecting Electronic Health Information, National Academy Press, Available at http://www.nap.edu/readingroom/books/for/

[15] J.G. Steiner, B.C. Neuman, and J.I. Schiller. "Kerberos: An Authentication Service for Open Network Systems," in Proceedings of the USENIX UNIX Security Symposium, USENIX Association, Berkeley, CA, August 1988