Evaluation of OpenID-Based Double-Factor Authentication for Preventing Session Hijacking in Web Applications

Asif Muhammad
School of Engineering and Technology, Asian Institute of Technology Bangkok, Thailand.
Email: Muhammad.asif@ait.ac.th

Nitin Tripathi
School of Engineering and Technology, Asian Institute of Technology Bangkok, Thailand.
Email: nitinkt@ait.ac.th

***Abstract*—Web users often find it difficult to manage their identities (IDs) due to large number of web applications. An effective and convenient ID management system is needed to handle the problem. OpenID is one of the better solutions to manage this task on heterogeneous web applications due to its lightweight and simple protocol. However, it is quite vulnerable to session hijacking, resulting in identity theft of a particular user. In this paper, we present a modified approach, based on double authentication that minimizes the risk of session hijacking in an OpenID environment.**

***Index Terms*—OpenID, PIN, Session Hijacking, Internet security.**

I. Introduction

With the exponential growth of Web 2.0 technology, there is a drastic increase in the number of individual websites which require user's registration to communicate with their services/sites. Identity management has emerged as one of the important fields in information technology, especially information security [1]. It is a primary mechanism for access control. Appropriate identity needed by every user who wants to access banking account, ecommerce web sites, or a company resource.

Digital identities can by managed by different ways, typically they can be managed at operating system level or at application level [1]. Although there are many other places to manage identities. Three most commonly used operating systems environments are: Unix/Linux, Microsoft Windows and Mainframe. Identity management In UNIX/Linux is done using LDAP, NIS, RADIUS, Kerberos and a number of other mechanisms. Active Directory (AD) is the most commonly used mechanism in the Microsoft world, There are multiple mechanisms for identity management in mainframe Systems like RACF (Resource Access Control Facility). A number of commercial and open source products are also available for identity management despite of the operating systems based identity management solutions. The products provide facilities to manage user identities across multiple platforms and services such as single sign on (SSO), cross company authentication (CCA), etc. Companies like RSA, Novell, Sun, and others provide commercial products for sophisticated identity management across multiple platforms. Most of the above mentioned identity management methods are used in two different ways, both for front-end user authentication and authorization as well as for back-end systems. Typically these systems work very well in a closed environment where all applications and systems are managed by a single company [1].

With the rapid increase in web-based systems over the internet caused a number of private and commercial websites. Users have to maintain their account with these websites and this start point for problems. Now a user has to create identities for all of the web sites and remember username and passwords. Obviously, this has created a number of issues not only for users but from security perspective as well [1].

The current circumstance for authentication of users seems unsustainable due to lot of security threats in today's world of web. Consequently, the users may have a lot of username and passwords to use for these different services and to keep this login information, users either have to note this information or use the same username/password for all the services. This is probably not acceptable because having the same username/passwords for all the services increases the probability of attack and a compromised website can lead to a highly undesired intrusion to all the sites/services being used by this user.

To manage different accounts on different websites for a user, there is a need to overcome different problems related to identity management on these sites. For example, one of the problem for a user on internet may be how one can manage different identities on different sites to which user needs to communicate with. The idea of OpenID is a good solution to this problem.

OpenID is a protocol which helps user to use URL as their identity across the OpenID enabled websites.

© 2012 ACADEMY PUBLISHER
doi:10.4304/jcp.7.11.2623-2628

Consumer websites can use this URL for authenticating the users. It is a new concept and allows the control of identity to users. Using this protocol, users have the liberty to decide which information should be sent to the consumer website for authentication purpose.

The OpenID provides *Single Sign-On* (SSO) service that a user can be authenticated in several web sites by submitting the password of OpenID to authentication server once. In this paper, we report an improved protocol using double-factor authentication that uses PIN code additionally to verify the credentials of the user. The results of case study are discussed in later sections.

II. Web Authentication Security

This section briefly describes the types of attacks that web users most often face when performing online authentication and how current HTTP security features address them.

Desktop compromise: a surprisingly large number of Desktop computers are compromised with malware [2]. Users of these compromised machines have zero guarantee of any security: all security indicators may be faked, and all host names may be hijacked. SSL is useless. Damage from these attacks is significant; though carrying out such an attack is typically more involved than either passive sniffing or social engineering [3].

Social engineering: users are easily fooled by malicious sites that visually spoof legitimate sites to steal credentials. Generally target of these sites include financial institutions and the other e-commerce websites through which intruders may gain financial benefits. Users generally don't check the URL or even the SSL padlock of their connections [3]. The damage from these attacks is well documented and significant [4] and carrying out such an attack is fairly trivial. Pharming attack is the most advance type of attack in this category, where a domain name server (DNS) record or even an internet protocol (IP) address is spoofed to make user believe that he/she is visiting the correct site. This problem may be somewhat alleviated with Internet Explorer 7's strong disincentive to visit inconsistent SSL sites. However, to our knowledge, there is no reliable data yet as to whether user behavior is significantly affected. This type of attack is on the rise via malicious open Wi-Fi base stations, which users tend to trust in their thirst for Internet access "on the go." Even when an incorrect SSL certificate raises a flag, users tend to ignore the warning [3].

Passive sniffing: It is common for general users to access web sites over open or insecure Wi-Fi access points, corporate proxies or un-switched local wired networks. The contents in response to their URLs request are easily sniffed able when SSL is not used. The damage from these kinds of attacks is unclear, as most non-SSL-using web sites are small providers. However, the threat is well understood: while the W3C does not mandate SSL, the W3C's technical advisory group is considering recommending that login credentials never be sent in the clear [5].

SSL is not enough: It is clear that SSL is not enough to protect against desktop compromise attacks. It is also relatively well understood that, for high-value applications, SSL is still not enough to protect against social engineering attacks, as evidenced by the depressingly high success of such social engineering attacks. The key issue is that, even with SSL, the web remains treacherous: a momentary lapse in judgment and Alice may be tricked into thinking that two 'v's are actually a 'w' [3]. As a result, some suggest that
High-value sites resort to two-factor authentication, where at least one factor is not easily stolen from an inattentive user.

III. OpenID

A. Benefits

1. The users can login to OpenID enabled websites without giving information about username and passwords to relying party or client website.
2. Users can control the information sending to the requesting websites depending upon the needs and the risk level.
3. It can be an alternative to SSO within an organization for different programs and applications.
4. For cross company authentication (CCA), it is an alternate to implement the concept for different entities involved in the system.
5. Reduces the cost of identity management and implementation.

B. OpenID Components

OpenID is a set of communication protocols. There are different parties involved to complete the cycle of this communication protocol. A normal set of communication messages is shown in figure-1among different parties of OpenID. The concept of OpenID is; users store their credentials on a central server that is called ID provider, server issues an ID in the form of URL to the user after collecting the user credentials, figure-1 shows how a user get access to an OpenID enabled website and is explained as follows

1. Users request the relying party or consumer website for getting access to the website by providing his/her ID.
2. In the second step, relying party communicates with the ID provider server to check authenticity of user.
3. In the third step, ID provider server redirects to user to ask the credentials of this user.
4. Once the user is authenticated by the server, server sends a protocol message to the requesting consumer website about the success/failure of this identity user.
5. Upon receiving success from server relying party give access to the user and starts communication.

© 2012 ACADEMY PUBLISHER

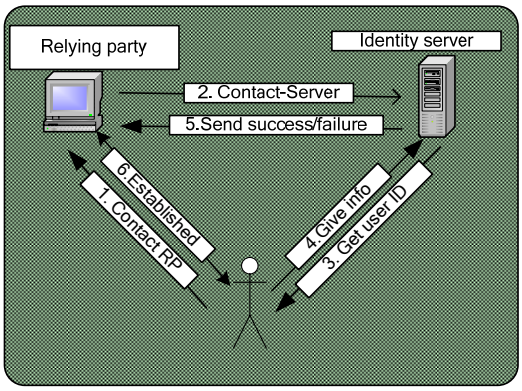

Figure 1. Normal Communication in OpenID environment.

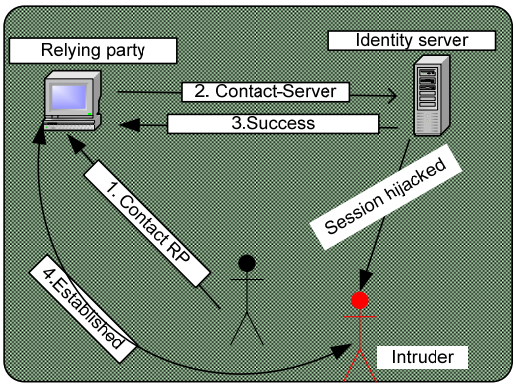

Figure 2. Session hijacking situation in OpenID environment.

IV. Session Hijacking Vulnerability in OpenID

The process of authenticating using OpenID has few security problems e.g. phishing attacks [6-9], also in the process of SSO in OpenID [11]. During the process of SSO in OpenID, the session information can be hijacked even through Secured Sockets Layer SSL [12]. In this paper OpenID 2.0 standard have been tested and used. Another experiment with standard 1.0 was done and that experiment showed the same vulnerability issue with session hijacking.

Session hijacking is a method in which an attacker or intruder steals the session ID of a user from the communication server or website. In this mechanism, the intruder or hijacker takes the session ID, and can act like an authorized user. Once the session is compromised then intruder can do anything like a normal user can do [6].

In OpenID environment, to prevent the session hijacking attack, it needs to use multi-factor authentication [11]. When the user starts communication with the relying party through his ID, and after confirmation of this ID from the server, relying party gives access to the user for communication. In step 3 of figure 1, if the session between user and server is compromised by an intruder and the intruder tries to access any OpenID- enabled website, then ID server directly gives positive response to relying party, and hence the relying party permits the intruder to get illegal access on OpenID. This problem is addressed with the help of two-factor authentication in the proposed system, figure 2 depicts situation of session hijacking in OpenID environment.

V. Methodology

Double authentication scheme is used to implement the prototype system. This technique uses two independent credentials for authenticating the user namely

1. User information which is stored on the ID server.
2. A PIN (Personal Identification Number) code.

In this prototype, Yet another distributed identity system (Yadis) protocol [9] is used for service discovery. Consumer website locates the OpenID server through this protocol.

OpenID components use different type of associate messages for successful communication to occur. Figure 3 shows a sample association message between the relying party and ID provider server. In this message second parameter "openid.session_type=DHSHA1" shows that we will exchange messages using Diffie-Hellman algorithm [7].

```
openid.mode=associate&openid.assoc_type=HMAC-SHA1&openid.session_type=DH-
SHA1&openid.dh_consumer_public=KC6IpA00A6S1CikafFS1rTGq19H8+de6GFi5YLKz4p
yDxUMS5Z8pMOm/Ptr1gFmCcgAXjFbuxS73ZutDTFJYpADoIntFVrah9eaezMcw6SDR24cnFjN
c14xq0zGt3QcRLXaNTRVKfMW8evDAmLCrvEhU5c7B3eqmk+bMMrbQpcE=&openid.dh_modul
us=ANz5OguIOXLsDhmYmsWizjEOHTdxfo2Vcbt2I3MYZuYe91ouJ4mLBX+YkcLiemOcPym2CB
RYHNOyyjmG0mg3BVd9RcLn5S3IHHoXGHblzqdLFEi/368Ygo79JRnxTkXjgmY0rxlJ5bU1zIK
aSDuKdiI+XUkKJX8Fvf8W8vsixYOr&openid.dh_gen=Ag==
```

Figure 3. Association message between relying party and ID provider server.

When the user requests the prototype system by providing his/her ID and PIN. This request is sent to the ID provider server by locating through Yadis service discovery protocol [9]. Upon receiving this request, the server asks for the credentials from the requesting user credentials for this ID. If user is authenticated, then server sends the PIN from the user credentials to the prototype system. Finally, the system gives authorization to the user after verifying the PIN code from server and user. This situation is shown in figure 4.

© 2012 ACADEMY PUBLISHER

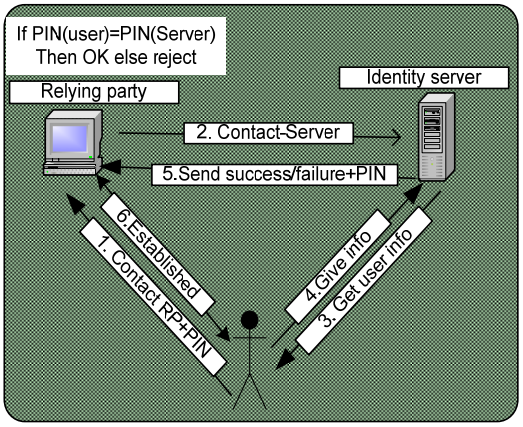

Figure 4. Communication using double authentication.

VI. Comparison with BeamAuth

Web-based authentication is vulnerable to a staggering number of social engineering attacks, typically called phishing [4]. Generally a spoofed page is presented to the user despite of the original page. Intruder can easily get user credentials through this fake page. The spoof can take the form of a simple user-interface deception, sometimes with a URL crafted to resemble the purported destination in order to trick even users who check the address bar. Recent variants, called pharming attacks [14, 15], are significantly more cunning: by spoofing DNS or even IP addresses the attacker's phishing URL matches exactly the purported destination. Pharming attacks are becoming easier to carry out using, for example, malicious base stations to which Wi-Fi users might innocently connect. The only remaining defense is the SSL certificate warning, which many users ignore [3].

Much recent work proposes defenses against phishing attacks, including site-specific password pre-processing [17], cryptographic protocols combined with trusted-path user interface indicators [3], and altogether novel methods of web authentication [18]. Unfortunately, all of these solutions require new client-side code, which greatly limits their deploy ability until major web browsers implement the feature and a large portion of web users upgrade accordingly. When the proposed change is implemented as a browser add-on, new trust and attack surface issues arise: the add-on usually has full control over the user's browser.

At a high level, it is well known that multi-factor authentication is preferable, though not foolproof, in defending against social engineering attacks. Yet multi-factor authentication is difficult to implement in an out-of-the-box browser. One extension-free approach to web-based two factor authentication is site-image verification, e.g. BankOfAmerica's Site Key [19] or Yahoo's sign-in seal [20]: the server provides a personalized login image to browsers previously tagged with a long-lasting cookie, and the user is expected to enter her password only if she notices her expected personal login image. The long-lasting cookie plays the role of a second factor, and the login image provides some form of human authentication of the server requesting the users Credentials.

BeamAuth provides second-factor authentication using a specially crafted bookmark instead of a cookie. They [10] believe this approach provides a few notable advantages:

1. BeamAuth token is hidden inside a bookmark rather than a cookie so that it is less vulnerable to cross-site scripting (XSS) attacks [21].
2. A bookmark has fewer privacy side-effects than a cookie, making it less likely to be deleted by routine cookie deletion.
3. A user's multiple browsers and computers can be automatically set up for BeamAuth using any one of numerous existing bookmark synchronization tools.

They [10] aim to make it more difficult to carry out social engineering attacks against customers of high-value web sites. High-value web sites should have an easy and relatively secure way to implement two-factor authentication without resorting to browser plug-in or physical tokens. They specifically aim to provide a "safety net" for users, so that a moment of inattention will not immediately result in identity theft. In other words, BeamAuth attempting to make phishing significantly more difficult for the attacker. Importantly, there aim is not to interfere with other proposals that may help address sophisticated pharming attacks.

Considering the high-value web sites, including in particular the single-sign-on use case in its many forms, where Alice is sent to her login page by a third-party web site, sometimes called the relying party because it relies on an authentication process performed by another party. For example, Flicker sends its users to Yahoo for authentication, and any Web application can use Yahoo in the same way with Yahoo BBauth [22]. A growing number of web applications use OpenID [23] for authentication, where the relying party is expected to redirect Alice to her OpenID server. A number of university networks also use this same technique: Harvard University's PIN system [24] and Stanford's Web Login system [25] are two prominent examples, where peripheral sites send users to the central login site which, after authentication, redirects the users back to the peripheral site with an authentication token. In all of these cases, phishing is of great concern, since Alice is sent to her login page by the site requesting authentication. It has been noted in particular that OpenID may make phishing easier because Alice explicitly discloses her identity provider, and thus the identity provider's look-and feel, to a potentially evil site [7]. BeamAuth aim to mitigate phishing attacks in this widespread scenario. In this paper the protocol is designed and experimented for preventing the session hijacking between the end use and the OpenID provider server. So the main difference between the proposed

© 2012 ACADEMY PUBLISHER

protocol in this paper and the beam auth is to handle different problems.

VII. Experimental Results

A social bookmarking prototype system is implemented in this study that provides an insight of the behavior of the proposed model. A comparison is made between our prototype system and one of most popular existing OpenID based consumer website [16], the screen shot of main window of LiveJournal is shown in figure 5.

This OpenID-based consumer website allows its users to enter the identity of user, and through its service discovery protocol, they do discover the location of the OpenID provider server, and ask for the credentials of this user if that user is not logged in to the server, then the client browser is redirected towards the server and ask for user credentials. After that it sends authentication message to consumer website in this case LiveJournal, and hence communication between the user and LiveJournal starts.

If the user already had a session with the identity server, then, the server redirects the user towards the LiveJournal site, and gets access to it without giving any information. So, in this case if the situation happens as discussed above, i.e., the session between client and the identity server is hacked by some intruder, then, the intruder can easily use this personal identity to any of the consumer website, without any authentication.

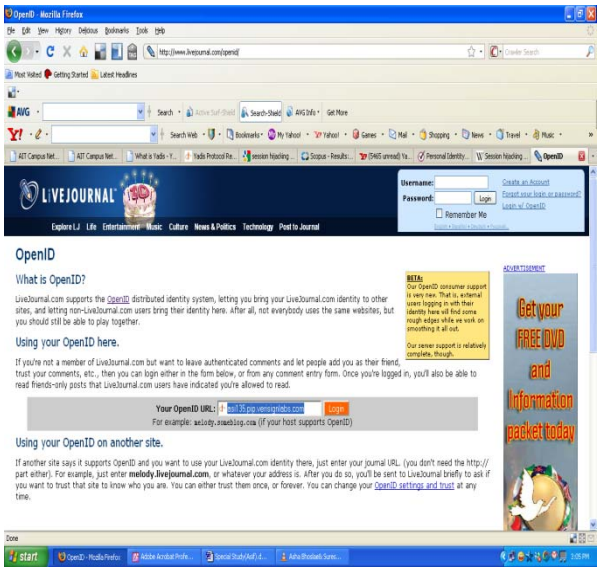

Figure 5. An OpenID-based consumer website.

Now consider the above case in our proposed model of communication between parties involved in OpenID communication, i.e., client, the consumer website, and the server. Every time when the user is trying to get access to a consumer website, then the procedure is as follows. User provides the personal identity, and a PIN code to the consumer website, in the case of our prototype system, this PIN code is already stored at the server by the user during initial registration with ID server.

Now, in the next step, consumer website requests to the identity provider server for the credentials of the user by approaching it through its discovery protocol. In this scenario, the server will send the PIN code from the user credentials that will be compared at the client side.

A screen shot of prototype implemented in this study is shown in figure 6.

Figure 6. The proposed secure protocol using OpenID.

VIII. Conclusion

Creating and managing a large number of accounts on different web applications, and services have become a challenge due to dramatic increase in the web traffic. OpenID is one of a good solution for identity management on web application services. However, OpenID is vulnerable to many types of threats such as phishing attacks and session hijacking. In this paper, we present, and evaluate an implementation of OpenID protocol, in order to provide a double factor authentication to the ID of a user. In the case study described, we compare the proposed approach to an existing OpenID provider service, which is LiveJournal.

The proposed model has the capability to stop malicious users to get illegal access to the consumer website as in the case of our prototype system. In summary, we conclude that even if the session between the user and the identity provider server is hijacked by some intruder, the intruder may not be able to get access to PIN code due to double factor authentication system.

Acknowledgement

I am heartily thankful to my supervisor, Dr. N.K. Tripathi, whose encouragement, guidance and support from the initial to the final level enabled me to develop an understanding of the subject. I am also thankful to my scholarship donor The Higher education commission who supported me financially for my studies.

Lastly, I offer my regards and blessings to all of my friends who supported me in any respect during the completion of the paper.

© 2012 ACADEMY PUBLISHER

REFERENCES

[1] Rafeeq Ur Rehman, *The OpenID book, A comprehensive guide to OpenID protocol and running OpenID enabled web sites*. 2008, Conformix Technologies Inc.

[2] Brian Krebs. Microsoft Releases Windows Malware Stats, June 2006. http://blog.washingtonpost.com/securityfix/2006/06/microsoft_releases_malware_sta.html. Last accessed on February 23rd. 2010.

[3] Rachna Dhamija, Doug Tygar, and Marti Hearst. Why Phishing Works. *In CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 581–590. ACM Special Interest Group on Computer-Human Interaction, January 2006.

[4] Anti-Phishing Working Group. Phishing Activity Trends, November 2006. http://www.antiphishing.org/reports/apwg_report_november_2006.pdf. Last accessed on February 28. 2010.

[5] Ed Rice. Passwords in the Clear, 2006. Available online at http://www.w3.org/2001/tag/doc/passwordsInTheClear-52, last viewed on February March 3rd, 2010.

[6] A web definition of session hijacking, available online at http://searchsoftwarequality.techtarget.com/sDefinition/0,,sid92_gci1188680,00.html. Last accessed on April 10th. 2010.

[7] Ben Laurie. OpenID: Phishing Heaven http://www.links.org/?p=187, Last accessed on February 3rd. 2007.

[8] OpenID Phishing Brainstorm, available online at http://wiki.openid.net/OpenID_Phishing_Brainstorm, last viewed on Januray 23rd 2011.

[9] Yadis protocol Specification. Available online at http://yadis.org/papers/yadis-v1.0.pdf. Last accessed on February 23rd. 2010

[10] BeamAuth: Two-Factor Web Authentication with a Bookmark: *In CCS 2007, Proceedings of the Fourteenth ACM Conference on Computer and Communications Security* (October 2007).

[11] Hyun-Kyung Oh, Seung-Hun Jin. The security Limitations of SSO in OpenID, *International Conference on Advanced Communication Technology, ICAT3*, art NO 4494089, pp.1608-1611.

[12] Thawatchai Chomsiri, HTTPS Hacking protection. *21st International Conference on Advanced Information Networking and applications workshops (AINAW'07)*, 2007.

[13] Two factor authentication definition from Wikipedia, available online at http://en.wikipedia.org/wiki/Two-factor_authentication. Last accessed on March 23rd. 2011.

[14] Gunter Ollmann. The Pharming Guide, available online at http://www.ngssoftware.com/papers/ThePharmingGuide.pdf. Last accessed on March 25th. 2011.

[15] V. Ramasubramanian and E. Sirer. Perils of transitive trust in the domain name system. *In Proceedings of the 2005 Internet Measurement Conference (IMC 2005)*, Berkeley, CA, USA, 2005.

[16] An Openid based consumer website. The live journal OpenID Enabled website, available online at http://www.livejournal.com/openid/. Last accessed on April 12, 2011.

[17] Blake Ross, Collin Jackson, Nicholas Miyake, Dan Boneh, and John C. Mitchell. Stronger Password Authentication Using Browser Extensions. *In P. McDaniel, editor, 14th USENIX Security Symposium*, 2005.

[18] Kim Cameron and Michael B. Jones. Design Rationale behind the Identity Meta system Architecture, 2006. http://www.identityblog.com/wp-content/resources/design_rationale.pdf. Last accessed on April 13th. 2011.

[19] Bank of America. Site Key. http://www.bankofamerica.com/privacy/sitekey/. Last accessed on December 13th. 2010.

[20] Yahoo. What is a sign-in Seal? http://security.yahoo.com/article.html?aid=2006102507, last viewed on 8 May 2007. Last accessed on December 25th. 2010.

[21] CERT Advisory CA-2000-02 Malicious HTML Tags Embedded in Client Web Requests. http://www.cert.org/advisories/CA-2000-02.html. Last accessed on January 23rd. 2011

[22] Yahoo. Browser-Based Authentication. http://developer.yahoo.com/auth/, Last accessed on January 23rd. 2011.

[23] D. Recordon and B. Fitzpatrick. OpenID Authentication 1.1 May 2006. http://openid.net/specs/openid-authentication-1_1.html. Last accessed on February 23rd. 2011.

[24] Harvard University. Harvard University PIN System. Available online at http:://pinharvard.edu/. Last accessed on February 3rd. 2007.

[25] Stanford University. Stanford Web Auth. http://www.stanford.edu/services/webauth/, last accessed on January 3rd 2011.

[26] Internet Engineering task force document , available online at http://www.ietf.org/rfc/rfc2631.txt. Last accessed on December 15th. 2010.

Asif Muhammad is a PhD candidate in computer science and information management program, school of engineering and technology, Asian Institute of Technology Bangkok Thailand. He received his master degree in computer science from Quaid-I-Azam University Islamabad, Pakistan.

He worked as a Software Engineer in a project of Air traffic control system for Pakistan air force from May 2006 to July 2007. After wards He joined the Asian Institute of Technology for his masters and PhD studies on a scholarship of Govt. of Pakistan. He was selected for a exchange program with National Institute of Informatics Japan where He carried out his part of research.

Nitin Tripathi is Associate professor and the coordinator of his department. He obtained his PhD from Indian institute of Technology, India. He obtained his M. Tech. in Remote Sensing from Indian Institute of Technology, Kanpur, India.

He is working as an Associate Professor in School of Engineering and Technology, Asian Institute of Technology Bangkok, Thailand.

Dr. Tripathi is Editor in chief of International Journal of Geo Informatics and an author of several Journal articles and conferences. Dr. Tripathi has published a number of books in his field.

© 2012 ACADEMY PUBLISHER