A Review Multi-Token Authorization Strategy For Secure Mobile Cloud Computing

Dr.Venkatakoti Reddy.G[1], B.V.Ramana[2], Dr. P.Bhaskara Reddy[3]
[1, 2]Assistant Professor, Dept of Computer Science and Engineering
[3]Professor, Dept of Computer Science and Engineering
[1, 2, 3] HITS, TS, India

Abstract- *cloud computing is an emerging paradigm shifting the shape of computing models from being a technology to a utility. however, security, privacy and trust are amongst the issues that can subvert the benefits and hence wide deployment of cloud computing. with the introduction of omnipresent mobile-based clients, the ubiquity of the model increases, suggesting a still higher integration in life. nonetheless, the security issues rise to a higher degree as well. the constrained input methods for credentials and the vulnerable wireless communication links are among factors giving rise to serious security issues. to strengthen the access control of cloud resources, organizations now commonly acquire identity management systems (idm). this paper presents that the most popular idm, namely oauth, working in scope of mobile cloud computing has many weaknesses in authorization architecture. in particular, authors find two major issues in current idm. first, if the idm system is compromised through malicious code, it allows a hacker to get authorization of all the protected resources hosted on a cloud. second, all the communication links among client, cloud and idm carries complete authorization token, that can allow hacker, through traffic interception at any communication link, an illegitimate access of protected resources. we also suggest a solution to the reported problems, and justify our arguments with experimentation and mathematical modeling.*

Keywords- Cloud Computing Security, Mobile Cloud Computing, Identity Management System. Secure Mobile Computing, Modified Identity Management System

I. INTRODUCTION

Over the past decade, enterprise computing has been shifting to a new paradigm, namely the cloud computing.

The cloud-computing paradigm provides several service models fitting the needs of an individual or organization. The ease of deployment, reduced costs, availability, scalability, accessibility, flexibility and location independence are some of the very strengths of this paradigm, giving rise to its popularity. On the contrary, security and privacy issues are limiting its wide spread deployment. Organizations are hesitant to storing and communicating valuable enterprise information to a third party outside their premises. In particular, the threat of unauthorized access to cloud data is of great concern, prompting researchers to propose novel authentication mechanisms. One such method is the deployment of a centralized Identity Management System.

Another emerging trend in enterprise computing is the use of smartphone devices. International Data Corporation reports on 33% increment trend on the sale of smartphones during past few years, with a prediction of 32.7% increase in 2013 [1]. Smartphone devices has been advanced greatly, in recent years, so has malicious code [2]. Although, smartphones are advancing in terms of computational power, rapidly replacing

Personal Computers (PCs) as first choice of a computing device [2], Nonetheless, their major problem still is that of resource poverty. To cater with this problem, organizations have started providing access to cloud services for their users with smartphone-based clients [3][4]. The location independence and computing power of a cloud joined with the mobility of a smartphone gives the freedom of computing anything anywhere, resulting in a powerful ubiquitous computing model. This power and flexibility is bringing high popularity to what researchers call Mobile Cloud Computing (MCC) [5][6]. ABI Research estimates that MCC will gain a user-base of 240 million by the end of 2015[7].

Being very convenient and accessible, smartphones are at a higher security risk than competing devices. This risk is mainly because of inherent nature of their application software and communication mechanism [2], as we explain further. First, tiny applications are easy to build by anyone, thus freely available, and hence contain malicious code in several instances. Second, mobile software development life cycle does not provide any activities ensuring the security, safety and trust. Third, the constrained resources do not allow executing full antivirus software. Fourth, the inherent nature of wireless links available to eavesdropping, and the wider

availability of Internet, even out of enterprise perimeter to access enterprise data leaves valuable information asset on risk. Fifth, the mobile users choose relatively simpler passwords that are easy to type with constrained input methods [5].

For all aforementioned reasons, strong authentication mechanisms for MCC are needed to protect privileged organizational data. In general, organizations deploy an IdM for greater access control, both for mobile based and PC-based client. However, our experiment, in this paper, shows that IdM based approach are not as effective for MCC as for conventional setting.

In this paper, we inspect security issues related to the use of smartphone-based clients acquiring cloud services through IdM. In particular, we discuss the problems of authorization for a protected cloud resource in two scenarios. First, when the organization's IdM is compromised through a malicious insider (i.e. malicious code) putting all the protected cloud resources on stake. Second, all the communication links among client, cloud and IdM carries complete authorization token, that can allow hacker, through traffic interception atany communication link, an illegitimate access of protected resources.

We proceed as follows. First, we discuss the background of our study in section II. In section III, we illustrate the scenario of the problem domain. We present the related work and the search methodology in section IV and V respectively. In section VI, we present our proposed solution deduced from the grounded theory and the experiments that we illustrate in section VII. We discuss limitations and future work in section VIII Finally, we conclude the paper in last section.

II. BACKGROUND

An identity management system manages the identities of individuals by ensuring their integrity throughout their lifecycle. It also maintains the associated roles, access rights, authorizations, and privileges [8]. Modern IdM's provide extended features like Single Sign-On (SSO) and federated identity management [9]. The federation of identity refers to linking the attributes of a person's identity across multiple services, or even organizations. And, an SSO refers to using one access token across multiple service and/or organizations. Popular examples of such federated identities/SSOs are Microsoft and Google accounts allowing users to use multiple services, sometime across multiple organizations. Fig. 1 illustrates the communication sequence between a user and an IdM. Figure 1 represents the basic functionality of IdM consisting of 8 steps that includes 1) user login to IdM with his username and password, 2) user request to access cloud application/date, 3) cloud ask for token, 4) user request the token from IdM, 5) IdM generates the token and send it to user and cloud, 6) user send the token, received from IdM, to cloud to finalize the process of authentication, 7) cloud compares the token received from user and IdM. On successful comparison, cloud let user access the data or application. The centralized management of identities of an organization's workers provides a solution that seems reliable, secure, and easy to deploy. Industry is adopting this mechanism on a fast pace. The deployment of IdM takes two layers, one for authentication, and another for authorization. Several options exist for deploying these layers, for example OpenID [10], SAML [11] and OAuth [12]. In our work, we embark upon authorization problems associated with the deployment of OAuth that we discuss later in section III.

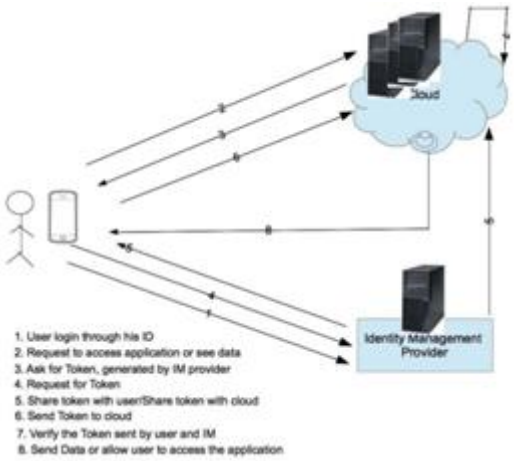

Fig. 1. Identity Management System

III. PROBLEMDOMAIN

We observed the limitations of IdM by looking into steps provided in Figure 1 such as what will happen if IdM is compromised. In step 4 and 5 (Fig. 1), IdM server generates the token and send it to cloud, and if IdM is compromised then any illegitimate user can use the same token to access the cloud's services/data. This compromise could be occurred due to malicious insider or malicious code. Current IdM, in case of being compromised, put all the cloud's resources on stake.

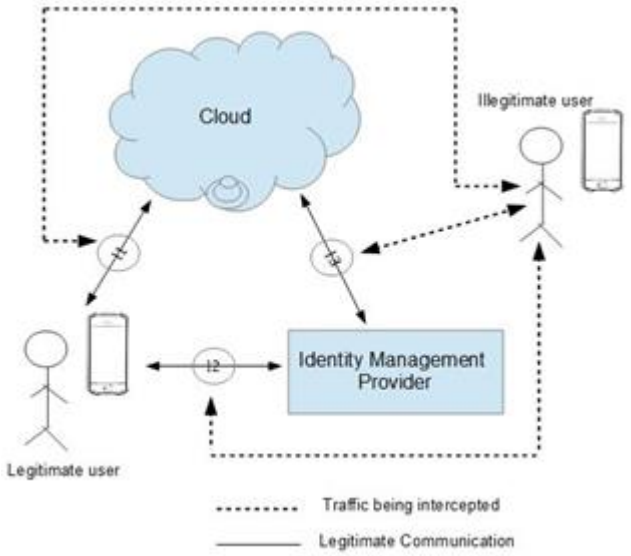

Fig. 2. Mediums and nature of traffic interception

Another problem, considered in this study, is what if an attacker intercepts the network traffic at any communication channel among IdM, cloud and user and gain unauthorized access to token that, further, can provide an unauthorized access to cloud's resources. We have three communication channels in our scenario, 1) between the mobile client and the cloud (marked with 1 in Fig. 2), 2) between the mobile client and the IdM (marked with 2 in Fig. 2), and 3) between the IdM and the cloud (marked with 3 in Fig. 2). Part of these communication links is obviously wireless, vulnerable to eavesdropping with very little effort.

If the hacker intercepts the traffic at Communication channel 1, he can access the token that is sent to cloud by user to access the data. Hacker can use this token to illegitimate access of cloud's resources. The same way, if he intercepts the traffic at communication channel 2 and 3, he will be able to get illegitimate access to token that is being used by user to access cloud's resources.

We assume, in this research, that this information over communication channel is not highly encrypted and can be decrypted with available decrypted algorithms.

For the aforementioned problems, we propose a solution in section VI. We do not work on strengthening the encryption on data link layer, nor do we suggest putting the best antivirus on OAuthserver. Instead, we propose a multi-token strategy that strengthens the IdM's authorization architecture within existing structure. It reduces the probability of theft of cloud data and service when IdM is compromised or network links are eavesdropped and tokens are stolen

IV. RELATEDWORK

OAuth [12] and OpenId [10] are two similar solutions that facilitate the idea of identity management systems. The purpose and approach to manage identities are different among these solutions.

In OAuth, client obtains a token (string denoting a specific scope and limited lifetime) from authorization server to access a resource, hosted on resource server. For example, end-user (resource owner) can grant printing service (client) access to her protected data, which is stored at data-storage-server (resource server) without sharing her credentials (username/password). OAuth consists of four modules (roles) that includes 1) resource owner (person/server that grant the access of a protected resource), 2) resource server (the server that hosts the protected resource), 3) client (user/application that make request to access resource on behalf of resource owner) and 4) authorization server (the server responsible to issue the token to client). Figure 3 represents the communication flow of OAuth and detail description is as follows [12]:

1. The client requests authorization from the resource owner
2. The client receives an authorization grant(credentials that represents the resource owner's authentication)
3. The client provides authorization grant to authorization server and request for access token
4. The authorization server authenticate the access token and after successful validation provides access token
5. The client request the protected resource from a resource server through by providing access token
6. The server validates the token and on successful validation, grants an access to resource.

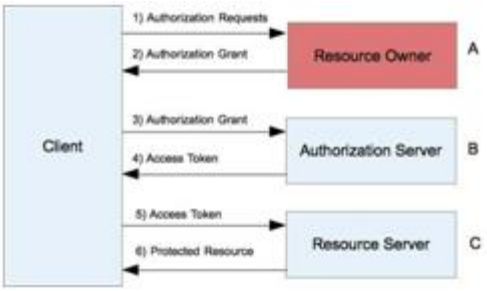

Fig. 3. OAuth communication protocol

OAuth considered, in official specification, a communication among modules (A, B and C in Figure 3) as out of scope. The specifications [12] for OAuth does not discuss the vulnerabilities and possible attacks that can be performed on the communication medium among these

modules. There is no discussion on the level of damage that can be caused to system, through intercepting the traffic among these modules. It is very important to secure each medium such as (1, 2, 3, 4, 5 and 6) so no information should be intercepted. Information leakage, at any medium in OAuth, can provide sufficient access to attacker to manipulate the resource/data. The assumption, made in OAuth specification [12], such as "attacker has no access to communication between authorization server and resource owner" reduce the implication of it in a secure system such as banking etc.

In addition to this limitation, Module A is a bottleneck of the system, if it is compromised through malicious insider or malicious code, the whole system would be compromised. The client, can access the information about authorization grant, and can access the token, that further can help to access the resource owner.

In case of smartphone being stolen, OAuth does not provide any mechanism to secure the data except that it encourages users to put key lock on their mobile [12].

Our research is based on authorization of user in the IdM and this is why OpenId is out of scope. OpenId is about authentication (providing the evidence who you are) but OAuth is about authorization (granting an access of resource to third party on your own behalf). OpenId helps you login in multiple sites through single sign-on. Studies such as [13][14][15] has proven that OpenId has many security weaknesses and vulnerable to malicious code attack. These studies discussed an attack, performed by attacker on server that uses OpenId, to install malicious code. This code forwarded the user to bogus identity provider authentication page and asked for his credentials. Later, attacker through malicious code used this credentials to access the user data on original server[14][15]. Many practitioners are promoting the use of OpenId with OAuth for better security. We observed that this combination of OAuth with OpenId could be lethal to user's private data. For example, in case of authorization server being compromised, OpenId (service for single sign-on) and OAuth (authorizing the person with single sign-on) could be an advantage to an attacker to access all resources/data of user on multiple sites.

Other than these two similar systems, there are many case studies that use IdM such as Xiao et al. in [16] mentioned current security mechanisms in mobile cloud computing as insufficient because if attacker is capable of faking/stealing user's credentials than the cloud data is on stake. Author in this study provides the algorithm to generate dynamic identities to provide secure mechanism to protect cloud data. This algorithm performs well if adequate security measurements are implemented at server level such as antivirus, network firewalling and intrusion detection systems. This algorithm has of no use, if the system is compromised, because whatever efficient key is generated through algorithm, attacker would get access to it. Leandro et al. in [9] promoted the use of Shibboleth (mechanism to control access) as access control system, in cloud computing, without the use of trusted third party e.g. IdM server. It provides strong authorization but does not provide strong authentication for example, once the user is authenticated, it does not provide a mechanism to ensure the legitimacy of the person connected with system whether a user is legitimate or an attacker.

In simple words, an illegitimate user holding valid username and password can access the cloud services without being verified. Shibboleth does not guarantee 100 % secure transaction. In order to deal with user verification, Angin et al. in [17] proposed a solution called 'active bundle scheme' for IdM with comparison of application-centric approach. This approach allows server to keep track of user in order to authenticate in such a way that does not reveal its actual identity and to protect personally identifiable information from unauthorized access. Authors in [9] discussed the similar concept except that Angin et al. do not implement or validate the solution. There are many articles such as [18][19][20][21][2] that provides IdM, with respect to PC, and modify it in order to secure user's data on cloud but we found no study, during our literature study, to implement IdM on mobile computing. We also observed that every article is modifying IdM just to protect user's identity, no one has explored it in the scenario where IdM server is compromised and network traffic is intercepted.

V. RESEARCH METHODOLOGY

Primarily, we use 2 research methods in this work. First, we conduct extensive literature survey (state-of-art). Based on the knowledge extracted from state-of-art and through empirical analysis, we trace problems in the current authorization architecture. The solutions to the problems—in the form of modified authorization architecture—are based on grounded theory established by extensive literature review of related work. To justify our solutions, we conduct experiments (state- of-practice) and do mathematical modeling in two scenarios, one for current practice in IdM and the other for suggested solutions. Finally, we compare the results of the two scenarios and show that our proposed solution provides better security.

VI. PROPOSED SOLUTION

Our experiments show that if IdM server is compromised, the attacker gets access to authorization token generated by IdM, resulting in an illegitimate access the protected resource on the cloud.

In our solution, we propose generating a distributed authorization token composed of two parts for a single resource access. First token is generated by IdM—upon producing credentials by the user—sent to the user and the cloud, as currently in practice. The second token is generated by the cloud—upon producing credentials by the user—and sent to the user. The cloud also saves this token for future use.

The sequence of action is as follows.

1. The user logs in to the cloud.
2. The cloud generates a token, sends it to the user and saves it as well for future reference. The cloud also requests the user to produce the token generated by IdM.
3. The user logs in toIdM.
4. The IdM generates the token and sends it to both the user and the cloud.
5. User sends both tokens—one from cloud and the other from IdM—to the cloud to request access.
6. The cloud compares the token sent by user with the tokens saved in its database.
7. The access is granted/denied on the basis of comparison results.

In this scenario, the cloud and the user possess two tokens, while the IdM server has access to a single token generated by it. It leaves a malicious insider planted into the IdM with access to insufficient information to acquire protected resources on the cloud. Fig. 4. Illustrates the communication architecture and sequence in our proposed solution.

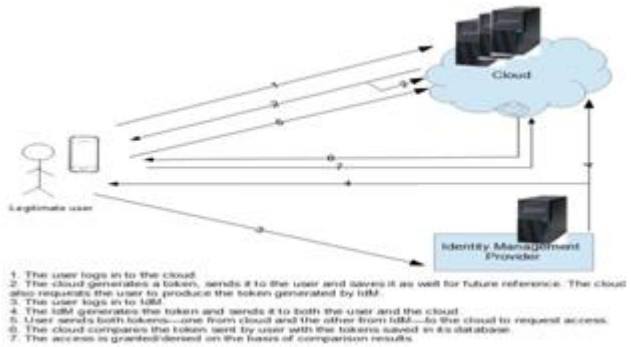

Fig. 4. Modified Identity Management System (Proposed Solution)

The client is required to login to the IdM as well as the cloud. Both servers respond with a token. Consequently, the user possesses two tokens, which are later represented to cloud to access the protected resource.

The other problem we report with the current authorization architecture is due to the insecurity of communication links, as discussed in section III. We assume that a hacker passively eavesdropping on communication links is able to read the communication, and strip off any security mechanisms applied, resulting in the recovery of the original token. She can then acquire the protected resource by presenting the token to the cloud. However, our proposed scenario limits the opportunity of a hacker to read sufficient information to acquire the protected resource.

In this scenario, if a hacker is eavesdropping on channel 2 or channel 3, as depicted in Fig. 2., she has access to only one token sent by the IdM to the user or the cloud. She is not ableto access the full information to acquire the protected resource. This significantly reduces the probability of hacking the required amount of information to acquire the protected resource. We analyze this situation in the following. Significant factor of 2 3. This section provides the discussion on our solution with respect to each problem, discussed in section III.

VII. PROTOTYPEIMPLEMENTATION/EXPERIMENT

We implement two scenarios in our experiment. First scenario is configured with the current authorization architecture of IdM. The other scenario deploys an IdM with our modified approach. We test the both scenarios with mobile clients of same specifications and configurations.

We use homogenous server systems to implement both IdM scenarios. The servers are installed with Xampp (Software Bundle), which provides Apache server 2.4.3, PHP 5.4.3 language support, along with MySQL 5.5.27 database management. We infect our sever for both scenarios with a malicious insider. Fig. 5. Represents the malicious code. In the first scenario of our experiment, whenever a user accesses the IdM by requesting an authorization token, her complete information is forwarded to a specified email address. The information comprises of username, password, token, and URL of the protected resource. The hacker consequently possesses sufficient information to access the protected resource.

VIII. LIMITATION AND FUTUREWORK

By the means of our experiments and mathematics, we produce the evidence that our proposed multi-token access strategy provides better security to the authorization token, and hence to the protected resources hosted on a cloud. A research limitation we find is in the question what if the cloud is compromised. However, it is another domain of research and researchers are working their ways to produce better cloud security models.

In the future, we want to work on a still better security model for MCC. Communication link l1 in our current model carries the complete token. A hacker eavesdropping on this link isable to acquire the protected resource. Contrary to this, the intended future model is expected to distribute the token in such a way that none of the communication channel carries the complete information about the authorization token. It shall leave the hacker with insufficient information to acquire the protected resource, in any case.

In general, mobile client users store their passwords into their browsers and applications. The behavior arises mainly due to the difficulty of typing with touch screen keyboards. These protected resources in case of device theft. We also plan to enhance our security model in a way that protects the resource from being accessed even when the mobile client is stolen, and the thief has access to stored credentials

Another issue is lacking of dynamic federation and agile mechanism in IdM systems [24] which is an architectural concern and should be addressed at design level.

IX. CONCLUSION

In the recent, third parties IdM's are introduced to manage digital identities and access control of the protected cloud resources an organization owns. The idea is similar to outsourcing the part of a project to some third party. Such systems are becoming very popular and commonly deployed in the organization especially for MCC clients. For MCC users facing mobile device's difficult input methods, IdM's popularity depends upon the ease of use. For the organization, their popularity is due to the reason that they allow organizations to use robust digital identity management systems without having a need to deploy one such system in their premises. However, research indicates some serious flaws into their access control model, like that of stealing the authorization tokens through a malicious insider or overa network link. We have worked our ways to identify those flaws empirically and with experimentation. As a solution, we propose some modifications—supported by experimentation— to the original access control model.

(UIC/ATC), 2012, pp. 627 –632.

M. Stihler, A. O. Santin, A. L. Marcon, and J. da Silva Fraga, "Integral Federated Identity Management for Cloud Computing," in *2012 5th International Conference on New Technologies, Mobility and Security (NTMS)*, 2012, pp. 1–5.

Primarily, we focus on distributing the authorization token generation between the IdM and the cloud. Through our experiments and analysis, we show that the possibility of hacking a token drops by a significant factor, resulting in increased security for the protected resource over the cloud.

REFERENCES

[1] "IDC, 'IDC Press Release', http://www.idc.com/getdoc.jsp?containerId=prUS22871611 (Access Date: 10 .9.2011)."

[2] I. Bernik and B. Markelj, "Blended threats to mobile devices on the rise," in *2012 International Conference on Information Society (i- Society)*, 2012, pp. 59–64.

[3] "R. G. Chicone, An Exploration of Security Implementations for Mobile Wireless Software Applications within Organizations. Minneapolis: Graduate Faculty of the School of Business and Technology Management, Northcentral University,2010."

[4] "M. K. Riedy, S. Beros, and H. J. Wen, 'Management Business Smart Phone Data' in Journal of Internet Law, pp.3-14,2011."

[5] F. Liu, P. Shu, H. Jin, L. Ding, J. Yu, D. Niu, and B. Li, "Gearing resource-poor mobile devices with powerful clouds: architectures, challenges, and applications," *IEEE Wireless Communications*, vol. 20, no. 3, pp. 14–22,2013.

[6] Z. Sanaei, S. Abolfazli, A. Gani, and R. Buyya, "Heterogeneity in mobile cloud computing: Taxonomy and open challenges," *IEEE Communications Surveys Tutorials*, vol. Early Access Online,2013.

[7] "https://www.abiresearch.com/research/product/1005283-mobile- cloud-applications/.".

[8] "Bishop, M., Computer Security: Art and Science, Reading, MA: Addison-Wesley Professional, 2002.,".

[9] M. A. P. Leandro, T. J. Nascimento, D. Santos, D. R, C. M. Westphall, and C. B. Westphall, "Multi-Tenancy Authorization System with Federated Identity for Cloud-Based Environments Using Shibboleth," presented at the ICN 2012, The Eleventh International Conference on Networks, 2012, pp.88–93.

[10] "http://openid.net/.".
[11] "http://saml.xml.org/.".
[12] D. H. <dick.hardt@gmail.com>, "The OAuth 2.0 Authorization Framework." [Online]. Available: http://tools.ietf.org/html/draft-ietf- oauth-v2-31. [Accessed:22-Jan-2013].
[13] R. Wang, S. Chen, and X. Wang, "Signing Me onto Your Accounts through Facebook and Google: A Traffic-Guided Security Study of Commercially Deployed Single-Sign-On Web Services," in *Proceedings of the 2012 IEEE Symposium on Security and Privacy*, Washington, DC, USA, 2012, pp.365–379.
[14] "OpenID still open to abuse." [Online]. Available: http://www.computing.co.uk/ctg/opinion/1824215/openid-abuse. [Accessed:22-Jan-2013].
[15] "http://lists.danga.com/pipermail/yadis/2005-June/000472.html.".
[16] S. Xiao and W. Gong, "Mobility Can Help: Protect User Identity with Dynamic Credential," in *2010 Eleventh International Conference on Mobile Data Management (MDM)*, 2010, pp. 378–380.
[17] P. Angin, B. Bhargava, R. Ranchal, N. Singh, M. Linderman, L. B. Othmane,andL.Lilien,"AnEntity-CentricApproachforPrivacyand
[18] Identity Management in Cloud Computing," 2010, pp. 177–183.
[19] Rosa Sanchez Guerrero, P. Arias Cabarcos, F. Almenares Mendoza, and D. Diaz-Sanchez, "Trust-aware federated IdM in consumer cloud computing," in *2012 IEEE International Conference on Consumer Electronics (ICCE)*, 2012, pp.53–54
[20] L.A.Martucci,A.Zuccato,B.Smeets,S.M.Habib,T.Johansson, and N. Shahmehri, "Privacy, Security and Trust in Cloud Computing: The Perspective of the Telecommunication Industry," in 2012 9th International Conference on Ubiquitous Intelligence Computing and 9th International Conference on Autonomic Trusted Computing
[21] P.Zhang,H.Sun,andZ.Yan,"BuildingupTrustedIdentity Management in Mobile Heterogeneous Environment," in *2011 IEEE 10th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, 2011, pp. 873 –877.
[22] I. Dacosta, S. Chakradeo, M. Ahamad, and P. Traynor, "One-time cookies: Preventing session hijacking attacks with stateless authentication tokens," *ACM Trans. Internet Technol.*, vol. 12, no. 1, pp. 1:1–1:24, Jul.2012.
[23] "http://www.apachefriends.org/en/xampp.html."
[24] R. Sanchez, F. Almenares, P. Arias, D. Diaz-Sanchez, and A. Marin, "Enhancing privacy and dynamic federation in IdM for consumer cloud computing," *IEEE Transactions on Consumer Electronics*, vol. 58, no. 1, pp. 95–103, Feb.2012.