Multi-factor web API security for securing Mobile Cloud

Conference Paper · August 2015

DOI: 10.1109/FSKD.2015.7382287

3 authors, including:

Liubo Ouyang
Hunan University
8 PUBLICATIONS **67** CITATIONS

SEE PROFILE

Wei-Tek Tsai
Arizona State University
491 PUBLICATIONS **10,229** CITATIONS

SEE PROFILE

Multi-Factor Web API Security for Securing Mobile Cloud

Longji Tang, Liubo Ouyang
College of Computer Science and Electronic Engineering
Hunan University
Changsha, China

Wei-Tek Tsai
Department of Computer Science and Engineering
Arizona State University,
Tempe, AZ 85287, USA

Abstract—**Mobile Cloud Computing is gaining more popularity in both mobile users and enterprises. With mobile-first becoming enterprise IT strategy and more enterprises exposing their business services to mobile cloud through Web API, the security of mobile cloud computing becomes a main concern and key successful factor as well. This paper shows the security challenges of mobile cloud computing and defines an end-to-end secure mobile cloud computing reference architecture. Then it shows Web API security is a key to the end-to-end security stack and specifies traditional API security mechanism and two multi-factor Web API security strategy and mechanism. Finally, it compares the security features provided by ten API gateway providers.**

Keywords- mobile cloud; security mechanism; web API; end-to-end

I. INTRODUCTION

Both cloud computing and mobile computing are the fieriest and evolving technologies in both IT world and business world. Mobile Cloud Computing (MCC) [32][10][6] is an architectural style with new emerging technology, which combines both cloud computing architectural style [31] and mobile computing architectural style [29] as well as both cloud infrastructure and mobile infrastructure to bring rich compute resources and services to mobile users, cloud service providers, and wireless network operators. MCC has been extending its domain from cloud to on-premise enterprises. The benefits for mobile both users and enterprises are (1) MCC enables execution of rich mobile applications on a plethora of mobile devices with rich user experiences. (2) MCC also leverages unified elastic resources provided by various cloud providers and their technologies as well as pay-as-you-go utility computing principle toward unrestricted functionality, storage, and mobility. (3) MCC provides new business opportunities for enterprises through exposing their services to mobile users. Therefore, Juniper Research predicates that mobile cloud computing will reach a market of $9.5 billion by 2014 [14].The ABI research predicted that enterprise MCC will be a $5.2 billion market by 2015 [2].

How does MCC make the connection between mobile applications and the services as well as data on- premise and in the cloud? How mobile application can be integrated with backend services? Modern Web API plays a key role in MCC architecture [32]. The Web API, specifically, the Representational State Transfer (REST) [7] API technology has been dramatically growth [30] and broadly adopted by enterprises due to needs from cloud computing and mobile computing, of cause, mobile cloud computing.

MCC is the combination of cloud computing and mobile computing, therefore, except for MCC itself challenges [20][26] , it also has challenges from both cloud and mobile computing[29][31] . Specifically, the security is a top concern for enterprises. A 2013 survey [23] shows 65% of organizations list security as their top concern in implementing cloud. Recently, iCloud's security issue increased security concern about data stored in the cloud. Our work in this paper defines an secure hybrid mobile cloud reference architecture with end-to-end security approach and mainly describes the Web API security technologies for securing mobile cloud.

This paper is organized as follows: Section 2 defines the end-to-end mobile cloud security through a secure mobile cloud reference architecture. Section 3 describes Web API security technologies. Section 4 provides a case study on Web Security application in mobile cloud computing. Section 5 discusses the related work; Section 6 summarizes conclusions of this paper.

II. MOBILE CLOUD WITH END-TO-END SECURITY

Securing mobile cloud is one of big challenges [20] for IT organizations adopting mobile cloud computing and gaining MCC benefits. In this section, we defined a secure hybrid mobile cloud reference architecture in which an end-to-end mobile cloud architecture is considered.

Figure 1 depicts the secure hybrid mobile cloud reference architecture in which the on-premise cloud backend services are included. **Error! Reference source not found.**I briefly describes major components in Figure 1.

TABLE I. DESCRIPTION OF COMPONENTS IN FIGURE 1

Components	Description
Mobile Devices	Smartphones, tablets, any other connected devices
MBaaS [32]	Mobile Backed as a Service which is kind of lightweight mobile cloud middleware
Web API Layer	The layer composed by SOAP Web services and REST Web services
Mobile-Enabled SOA Enterprises	The enterprises with SOA architecture support accessing their services and data

	from mobile devices
Cloud Service Providers	It includes SaaS, PaaS, and IaaS services providers, such as Google, Amazon, Saleforce.
Management	Management systems in mobile cloud computing
Security Stack	End-to-end security stack for mobile cloud computing

The MCC end-to-end security stack in Figure 1 is briefly described and discussed as follows:

- Device Security – it includes (1) the system and app security provided by Original equipment manufacturers (OEMs), such as iOS security from Apple; (2) device management, such as MDM, MAM, and BOYD [29][32] .
- MBaaS Security – MBaaS platform provides Identity, such as IDM and ACLs for user access authorization and authentication. It also provides platform security, such as firewall, full-sandboxing, strict employee access control for BOYD. It supports compliance with security auditing.
- API Security – it is playing more important role in the MCC, since Web API, specifically, the public API is becoming a layer between mobile applications and public cloud services as well as on-premise cloud services. In particular, more and more enterprises expose its core business assets to mobile devices running everywhere and everytime. API Security is becoming a very important layer in the end-to-end security stack. We will discuss API security in the section 3 and 4 in detail.
- Backend Service Security – it includes various security strategy and technologies, such as firewall, virtualization security for securing public clouds (SaaS, PaaS, and IaaS); Ping Identity, LDAP, IDM for protecting private clouds and on-premise services.
- Data Protection – It is the fundamental for all security layers, since the services serve mobile applications and users through accessing the data storing in the cloud or on-premise data storage. [20] discussed various data protection technologies.
- Policy and Standards – all security technologies are built on certain security policy and standards. The standards include security standards from ISO, NIST, RFC, and other IT standard organization, such as open authorization standard OAuth 2 was published RFC 6749 [12].

The end-to-end security stack is independent of any technologies and implementation chosen by MCC providers for building a MCC system. This paper will focus on Web API security for securing mobile cloud in the rest of sections.

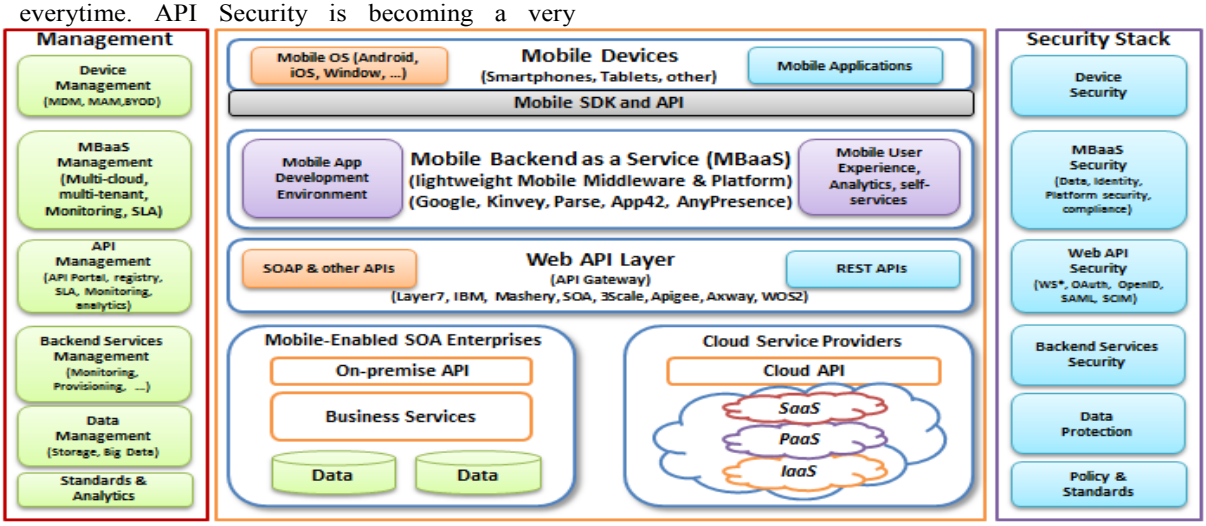

Figure 1. Secure Hybrid Mobile Cloud Reference Architecture

III. Web API SECURITY

Section 2 defined an end-to-end secure mobile cloud reference architecture. Our work in this paper will focus on the Web API security which is taken serious consideration now in era of APIs, where companies are making their core assets and business services available for others to become a crucial integrated ecosystem. In this era, security risk for intrusion attacks and data theft is evident. Recently, Twitter security breach that exposed more than 250,000 accounts [33]. [24] reported 42% Web API consumers encountered security issues. This section focuses on the solutions for Web API security.

Web API in this paper is defined as SOAP or REST web service which comes as just a specification of remote calls exposed to the API consumers. SOAP web service API has relevant mature security mechanism based on its WS-security standards [31]. However, in mobile cloud world, REST web service API has been becoming mainstream [5][30] due to its simplicity and ubiquity. The section will mainly discuss REST API security challenges, solutions, and standards.

Without specific notice, the API will indicate REST style Web API in the rest of paper.

Unlike SOAP-based Web API with complicated WS-* standards [31], the REST API is based on REST architectural style [7], which is using text-based JSON message over HTTP transport. Whereas WS-* have built in too much security, REST like HTTP and internet itself built in too little security. That is one big challenge for securing REST API, since the weak build-in security often leads web API vulnerability, such as injection attacks, sensitive data exposure, and incomplete access control. Open Web Application Security Project (OWASP) listed top 10 security vulnerabilities [22] in which half of them are related REST API vulnerabilities. Moreover, modern internal applications and services no longer run in isolated on-premise data center, they are now connected to partners, customers, and services outside the control of IT and application owners through web API, more than half of traffics by REST API. Specifically, the "internal" applications and services are exposed to mobile users from anywhere and anytime. Making those mobile cloud connectivity secure is critical for enterprises. Controlling and tracking access outside the firewall through web API and protecting the business assets on-premise and in the cloud is becoming a very challenge task for today's IT.

Web API security is a vast field with many different meanings and definitions. This paper starts from common attributes of security, and then specifies API security strategy and mechanism by using security factors. **Error! Reference source not found.** provides the definition of common security attributes.

TABLE II. DESCRIPTION OF COMMON SECURITY ATTRIBUTES

Security attributes	Description
Authentication	Reliable identify end user
Authorization	Give identified user to access to right resources, services, and data
Encryption	Hide information from unauthorized access
Signatures	Ensure information integrity
Vulnerability	Preventing attacks and damage to service consumers or providers

Not every API is the same and also not every API accesses the same services and data with same security requirements. This paper specifies API security into three levels based on (1) how secure the resources, services, and data are needed (2) the security factors:

Weak API Security: the API and its resources are only protected by traditional security mechanism which includes basic HTTP authentication and SSL:

- The HTTP/1.0 specification first defined the scheme for HTTP basic authentication. With this model, users must authenticate themselves with the corresponding username and password for each realm.
- Secure Socket Layer (SSL) is used to encrypt HTTP messages, sent and received either by web browsers or API clients.

The weak API security is kind of one-factor security and broadly adopted in mobile cloud applications. The mechanism is good for open free API with no data security requirement. For example, The GitHub REST API is protected with Weak API Security. Before you proceed any further, you need to create a GitHub account at https://github.com. However, weak API security does only protect the transport layer between API consumers and web servers, the data is not protected. Moreover, the transport layer security (TLS) vulnerability has been discovered. To protect API economy for API consumers as well as providers, and the services and data from enterprises and mobile cloud service providers, specifically when API is for business transaction processing with secure data request, weak API security is not good enough. Let us specify another two API security mechanisms as follows.

API Security: the API and its resources are protected by the following two-factor security mechanism:

- Traditional TLS security in weak API security
- Modern API security mechanism which include
  - Authentication: OAuth2 clientID, OpenID, API Key management, Identity Management, such as Ping IdM and CA IdM, Security Token Service (STS) [35]
  - Authorization: Token-based OAuth2, role-based or policy-based ACL
  - Sensitive data encryption
  - Security monitoring and attack prevention

Username and password pair based on weak API security is vulnerable, since the passwords have been stolen, cracked, phished, guessed, sniffed, captured, and leaked. Modern authentication methods are trying to reduce the vulnerability of traditional methods. If your apps just make API calls without accessing private user data, then a simple API key mechanism can be used as API authentication to identify the API user and the API consumer's apps for accounting and traffic purpose, as shown in Figure 2:

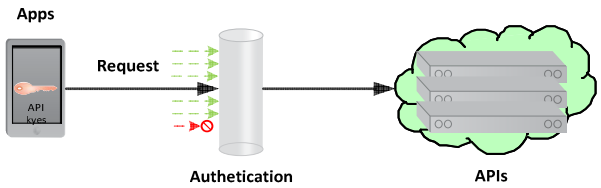

Figure 2. Simple API Key Authentication

If API consumers not only call API, but also access private data, then API key is not good enough to authenticate API users, the OAuth clientID can be for protecting this type of API access [9].

OpenID (OID) is an open authentication standard and decentralized protocol [21] that allows users to be authenticated by certain co-operating web sites using a third party service. With OID, users can log into multiple different web sites owned by different companies in Replying Parties group without registering with their information for each web site. OpenID Connect is the latest OpenID standard, which integrates OAuth2 as its authorization protocol.

To protect enterprises API, the user credential, such as user account, should be identified; therefore the IDM and LADP are adopted for authenticating users outside of enterprises and internal users, respectively.

OAuth2 is open authorization protocol and becomes a RFC standard rfc6749 [12], which creates a consistent, flexible, and policy-based authorization framework for web applications, RESTful web services, mobile devices, and browsers attempting to communicate with REST API, specifically the cloud API. Figure 3 shows that the OAuth2 clearly separates the role of authorization from access control by two different participants: authorization server and resource server. Therefore, users can use a central authorization server for accessing multiple resource servers. It is similar to traditional Single Sign-On (SSO) architecture. Moreover, the functionality of an OAuth2 authorization server is same as an API Security Token Service (STS).

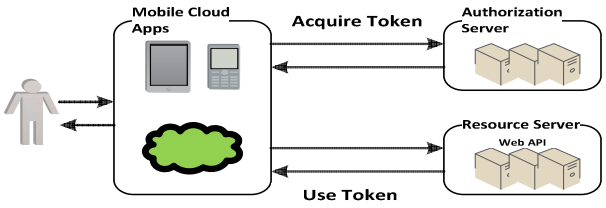

Figure 3. OAuth2 Authorization

The API security specified in the section is broadly adopted by most of mobile cloud service providers, such as Google, Amazon, IBM, Microsoft; API management providers, such as 3Scale, Layer 7, Merphy; Many enterprises, such as FedEx, WalMart. OAuth2 does not provide an user authentication protocol and just suggested client applications should use authentication for protecting the token endpoint, which gives client applications flexibility to choose their authentication mechanism. OAuth2 implementation may have more cost due to its complexity; therefore there is trade-off between the OAuth2 complexity and a simpler system just based on shared secrets and user account database. That is reason why some enterprises still favor simple SSO. The trend is modern security mechanism, such as OAuth2 will be adopted by enterprises with their API exposing to their partners and mobile cloud applications. Another alternative approach is to use Security Assertion Markup Language (SAML), an OASIS standard [25]. SAML and OAuth2 share a lot of goals, but SAML implementation is even more complicated, however by SAML specification, it is very secure and SAML can enable both web-based authentication and authorization. Therefore, SAML is adopted by major SaaS providers, like Google and Saleforce for their SAML 2.0 SSO which increases their security on using traditional SSO.

Recently, the vulnerability of OpenID and OAuth2 has been found [16], which could lead to open redirect attacks for both clients and security providers. The vulnerability affects large mobile cloud service providers, such as Facebook and Google. Moreover, enterprises have more strong security requirement for protecting their core business assets when more API exposing to their partners and mobile cloud applications. In the rest of section, we define a Strong API Security.

Strong API Security: the API and its resources are protected by the following three-factor security mechanism:

- Basic authentication which includes API user registration, SSL, and strong password protection
- Adopt modern standard-based security mechanism specified in ***API Security***. Integrate with enterprise IdM and LDAP. Provide REST JSON message level security, such as JSON Web Token (JWT), JSON Web Signature (JWS), and JSON Web Encryption (JWE) [27].
- Adopt security mechanism between API and its backend services as the third security factor.

The Strong API Security added the third security factor to API security for protecting the backend services and data accessed by API as shown in **Figure 4**:.

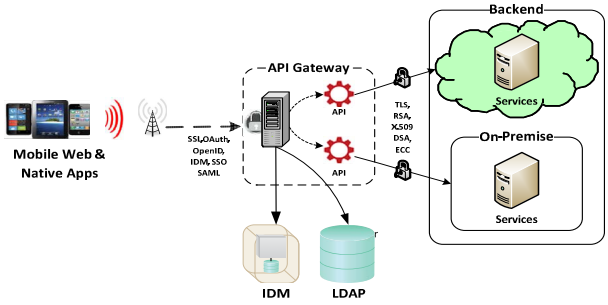

Figure 4. Strong API Security Architecture

The third security factor includes token-based STS API-to-Backend authentication, Public Key Infrastructure (PKI), additional TLS and data transmission security, such as X.509 PKI [12], RSA or DSA key, FCC crystal structure, TLS Handshake protocol [27], in which X.509 standard is broadly adopted by enterprises API-to-backend security and app-to-app security. The X.509 is an ITU Telecommunication Standardization Sector (ITU-T) standard in cryptography for a PKI and Privilege Management Infrastructure (PMI). X.509 specifies, amongst other things, standard formats for public key certificates, certificate revocation lists, attribute certificates, and a certification path validation algorithm. Figure 5 describes an API-to-Backend X.509-based certificate security architecture.

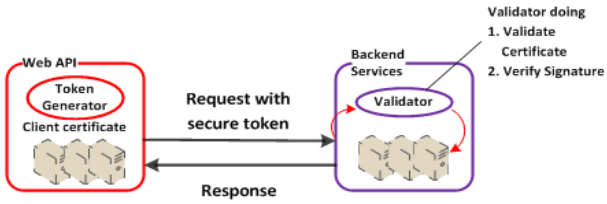

Figure 5. X.509 Certificate Security Architecture

In summary, the section specifies three types of API security models which are Weak API Security (One-factor security), API Security (Two-factor security), and Strong API Security

(Three-factor security). Every model has its use cases and its advantages as well as disadvantages. Figure 6 depicts a token-based three-factor strong API security model. Compared with other two models, it has the strangest security, but it is more complicated and has more security cost. In general, this kind of model is suitable for financial transaction processing, such as API credit card payment.

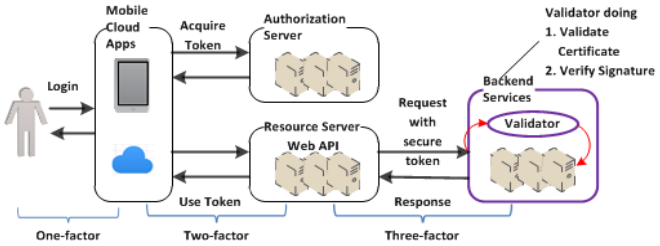

Figure 6. Three-Factor API Security Architecture

IV. COMPARISON OF API SECURITY PLATFORMS

This section compares API Security platform from ten API gateway providers. Industry API security platforms are provided by API gateway which is part of API management [30]. **Error! Reference source not found.** compares the main API security features and their pricing models provided by ten API gateway providers.

In **Error! Reference source not found.**, all API gateway providers support two-factor API security, most of them also support three-factor API security which is enterprise-strength security. Moreover, the XACML [36] is the short name of "eXtensible Access Control Markup Language", which is an OASIS language standard implemented in XML for defining declarative access control policy and a process model describing how to evaluate access requests based on the rules defined in policies. The standard promotes common terminology and interoperability between access control implementations by different vendors. Recently, REST profile and JSON profile described by XACML are developed, which is very helpful at describing access control policies of REST/JSON API security.

TABLE III. COMPARISION OF API SECURITY FEATURES OF TEN API GATEWAY PROVIDERS

API Gateway Provider	**API Security Features**								**Pricing model**
	Enterprise-strength security and access control (such as X.509 PKI, STS)	*Security Firewalling*	*Authentication, Authorization, and Audit*	*XACML*	*OAuth2*	*SAML*	*API Key Management*	*Threat Protection*	
IBM [11]	Y	Y	Y	Y	Y	Y	Y	Y	License
Google [9]	Y	Y	Y	Y	Y	Y	Y	Y	Pay-as-you-go
Microsoft [19]	Y	Y	Y	Y	Y	Y	Y	Y	Pay-as-you-go
Layer7[15]	Y	Y	Y	Y	Y	Y	Y	Y	License
SOA Software [28]	Y	Y	Y	Y	Y	Y	Y	Y	License
Mashery [18]	Y	N	Y	N	Y	Y	Y	Y	License
Apigee [3]	Y	Limited	Y	N	Y	Y	Y	Y	License
WSO2 [34]	Y	Limited	Y	Y	Y	Y	Y	Y	Free open source
Forum [8]	Y	Y	Y	N	Y	Y	N	Y	License
3Scale 0	Limited	N	Limited	N	Y	N	Y	Y	Basic free + subscription

V. RELATED WORKS

The work presented in this paper is rooted in the following two main research directions: (1) securing mobile cloud computing; (2) API security.

Mobile cloud computing has been introduced in [5][10][6][32] which discussed general security issues and solutions of mobile cloud computing. [20] is survey on securing mobile cloud computing which covers network security, data security, and cloud security in mobile cloud computing. Moreover, it described and compared various mobile cloud computing security framework. However, the important aspect - API security of mobile cloud computing is not discussed.

There are a lot of research literatures and books on SOAP based API web service security. Since this paper focuses on REST API, we will not list them. [30][32] discuss REST API security in general. [27] discussed various advanced API security technologies. Standard community, such as OASIS and IETF, published API security standards, such as OAuth2, XACML. API management providers, such as IBM, WOS2, Layer7, published a lot of white papers on API security. These API security researches do not cover the multi-factor API security for securing mobile cloud computing.

VI. CONCLUSIONS

This paper discussed the security challenges in mobile cloud computing, defines an end-to-end secure mobile cloud computing reference architecture, and mainly describes the multi-factor API security. The contributions of this paper are:

1) Defining an end-to-end secure mobile cloud computing reference architecture.

2) Specifying three types API security models with different strategies and mechanisms in mobile cloud computing.

3) Providing an informal comparison of the API security features provided by ten API gateway providers.

Our research work has been motivated by the desire to understand and evaluate the Web API security models, technologies, and standards for securing mobile cloud computing. Since Web API is a key component setting between mobile cloud applications running in devices and the services and data on-premise as well as in the cloud, API security becomes the key in security quality of mobile cloud computing. Our initial research on API security is helpful at further security research of mobile cloud computing.

VII. ACKNOWLEDGEMENT

This work was supported by Science and Technology Program of Hunan Province, China (2015GK3009).

REFERENCES

[1] 3Scale, API Management Platform, http://pages. 3scale. net. 2014
[2] ABI Research, Mobile Cloud Computing Reaches $5.2B Market By 2015, http://mobileenterprise.edgl.com/top-stories/ABI-Research--Mobile-Cloud-Computing-Reaches-$5-2B-Market-By-201560640,
[3] Apigee, API Management Platform, http://apigee. com. 2014
[4] A. Cecil Donald, S. Arul Oli, L. Arockiam, Mobile Cloud Security Issues and Challenges: A Perspective, International Journal of Engineering and Innovative Technology, Volume 3, Issue 1, July 2013.
[5] D. Chappell, D. Chou, T. Erl, F. Lascelles, M. Little, T. Rischbeck, A. Simon, R. Stoffers, and L. Tang, Service Infrastructure: On-Premise and in the Cloud, The Prentice Hall Service Technology Series from Thomas Erl, February 19, 2015
[6] N. Fernando, S. W. Loke, and W. Rahayu, Mobile cloud computing: A survey, Future Generation Computer Systems, 2012
[7] R. T. Fielding, Architectural Styles and the Design of Network-based Software Architectures, PhD Thesis, University of California, Irvine, 2000
[8] Forum Systems, API Management Platform, http://www.forumsys.com, 2014
[9] Google, API Keys, https://developers.google.com
[10] B. Hu, J. Wang, LJ Zhang, C. Xing, and R. Wang, A CCRA-Based Architecture for Enterprise Mobile Cloud Computing, MS '13 Proceedings of the 2013 IEEE Second International Conference on Mobile Services, P. 87-94, 2013.
[11] IBM, Exposing and Managing Enterprise Services with IBM API Management, Redbook, 2014
[12] IETF, The OAuth 2.0 Authorization Framework, https://tools.ietf.org/html/rfc6749, 2012
[13] IETF, Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile, http://tools.ietf.org/html/rfc5280, 2008
[14] Juniper Research, Cloud-based Mobile Market to Grow 88%, http://www.marketingcharts.com
[15] Layer7, API Management Platform, http://www.layer7tech.com/, 2015
[16] A. Low and S. Rosenblatt, Serious security flaw in OAuth, OpenID discovered, May 2, 2014, http://www.cnet.com/news/serious-security-flaw-in-oauth-and-openid-discovered
[17] E. Maler and J. s. Hammond, The Forrester Wave™: API Management Platforms, Q1 2013
[18] Mashery, API Management Platform, https://www.mashery.com
[19] Microsoft, API Management Platform, http://azure.microsoft.com/en-us/services/api-management
[20] A. Nasir Khana, M.L. Mat Kiaha, S. U. Khanb, and S. A. Madani, Towards secure mobile cloud computing: A survey, Future Generation Computer Systems, 2012
[21] OpenID Foundation, OpebID and OpenID Connect, http://openid.net/
[22] OWASP, Top 10 Security Vulnerabilities, 2013, https://www.owasp.org/index.php/Top_10_2013
[23] PC Connection, Overcoming the Security Challenges of the Cloud, 2013
[24] Parasoft, API Integrity: How Buggy Are Today's APIs, 2013
[25] SAML, http://en.wikipedia.org/wiki/SAML_2.0, 2013
[26] A. Shahzad and M. Hussain, Security Issues and Challenges of Mobile Cloud Computing, International Journal of Grid and Distributed Computing, Vol.6, No.6 (2013), pp.37-50
[27] P. Siriwardena, Advanced API Security, aprèss, 2014
[28] SOA Software, API Management Platform, http://soa.com/,2014
[29] L. Tang, Wei-Tek Tsai, and J. Dong, Enterprise Mobile Service Architecture: Challenges and Approaches, Service-Driven Approaches to Architecture and Enterprise Integration, IGI Global, 2013
[30] L. Tang and M. Little, API Governance and Management, October 8, 2014 , Service Technology Magazine Issue LXXXVI, http://servicetechmag.com/I86/0914-1,2013
[31] L. Tang, Modeling and Analyzing Service-Oriented Enterprise Architectural Styles, PhD thesis, UMI/ProQuest, 2011
[32] L. Tang, Mobile Service Infrastructure, in book [5]
[33] Twitter, Keeping our users secure, Feb. 1, 2013, https://blog.twitter.com/2013/keeping-our-users-secure.
[34] WOS2, API Management Platform, http://wso2.com/platforms,2014
[35] Wikipedia, Security token service (STS), http://en.wikipedia.org/wiki/Security_token_service,2013
[36] XACML, http://en.wikipedia.org/wiki/XACML, 2015