Choice of Suitable Identity and Access Management Standards for Mobile Computing and Communication

Nitin Naik[1], Paul Jenkins[1] and David Newell[2]

[1]Defence School of Communications and Information Systems, Ministry of Defence, United Kingdom
[2]Department of Computing and Informatics, Bournemouth University, United Kingdom
Email: {nitin.naik100, paul.jenkins683}@mod.gov.uk, dnewell@bournemouth.ac.uk

***Abstract*—Enterprises have recognised the importance of personal mobile devices for business and official use. Employees and consumers have been freely accessing resources and services from their principal organisation and partners' businesses on their mobile devices, to improve the efficiency and productivity of their businesses. This mobile computing-based business model has one major challenge, that of ascertaining and linking users' identities and access rights across business partners. The parent organisation owns all the confidential information about users but the collaborative organisation has to verify users' identities and access rights to allow access to their services and resources. This challenge involves resolving how to communicate users' identities to collaborative organisations without sending their confidential information. Several generic Identity and Access Management (IAM) standards have been proposed, and three have become established standards: Security Assertion Markup Language (SAML), Open Authentication (OAuth), and OpenID Connect (OIDC). Mobile computing and communication have some specific requirements and limitations; therefore, this paper evaluates these IAM standards to ascertain suitable IAM to protect mobile computing and communication. This evaluation is based on the three types of analyses: comparative analysis, suitability analysis and security vulnerability analysis of SAML, OAuth and OIDC.**

***Index Terms*—Mobile Computing and Communication; Identity and Access Management; IAM; SAML; OAuth; OpenID Connect; SSO**

I. Introduction

Mobile computing and communication enables enterprises and consumers the flexibility to conduct their work at a time and place which is convenient (anytime, anywhere), increasing productivity and efficiency. However, this flexibility of mobile devices in the enterprise presents additional security challenges [1]. The biggest issue is to securely authenticate and authorize staff and clients among all the collaborative organisations. The parent organisation owns all the confidential information about users but collaborative organisations have to verify users' identities and access rights to permit access to their services and resources. This challenge involves resolving how to transmit users' identities to collaborative organisations without sending their confidential information. The Identity and Access Management (IAM) standard can be used as an effective solution for authenticating and authorizing users across all collaborative organisations [1].

IAM standards have been developed to support all authentication and authorization activities at a corporate level. IAM service providers offer a wide range of activities such as user creation, user activation, user authentication, user authorization, user activity monitoring, application usage, user provisioning, user deprovisioning, user auditing, and user revocation [2], [3]. There are various IAM standards available, of which, the established and effective IAM standards are SAML, OAuth, and OIDC. These three IAM standards cover the majority of the IAM marketplace. Currently, identity management services became a stand-alone IT function known as IDaaS (Identity-as-a-Service). Mobile computing and communications have several challenges such as mobility, resource scarcity, heterogeneity, and insecure wireless mobile communication [4], [5]. To consider these limitations, the IAM standard for mobile computing requires some adaptations such as lightweight protocols, energy efficiency, native mobile app support and robust security [6], [7]. Therefore, this paper evaluates these IAM standards to ascertain suitable IAM to protect mobile computing and communications. Three types of analyses are examined: comparative analysis, suitability analysis and security vulnerability analysis of SAML, OAuth and OIDC.

The rest of the paper is organised as follows: Section II elucidates SAML, OAuth and OIDC use cases for mobile computing and communication; Section III expounds the comparative analysis of these IAM standards; Section IV critically evaluates the suitability of these IAM standards for mobile computing and communication; Section V performs security vulnerability analysis of IAM standards; Section VI concludes the paper and suggests some future work.

II. SAML, OAuth and OIDC Use Cases for Mobile Computing and Communication

A. Security Assertion Markup Language (SAML)

SAML is an XML-based framework developed by OASIS for communicating user's information related to authentication and authorization [8]. It permits the two federated partners

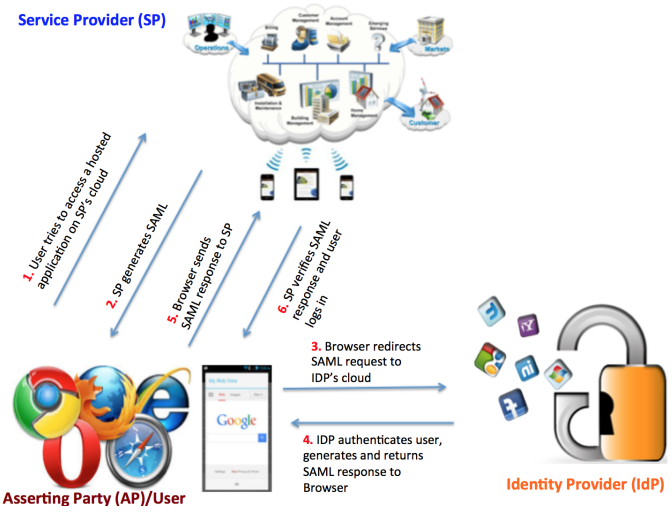

Fig. 1. A SAML Use Case for Mobile Computing and Communication

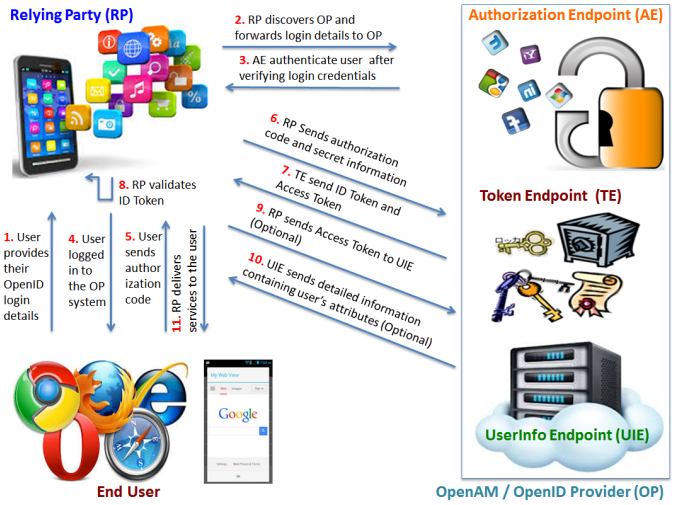

Fig. 3. An OIDC Use Case for Mobile Computing and Communication

to select and share necessary identity attributes they require in a SAML message/assertion provided that they can be represented in XML [7]. A typical use case of SAML for mobile computing and communication and its corresponding steps are illustrated in Fig. 1 [2].

B. Open Authorization (OAuth)

OAuth is mainly an authorization protocol. OAuth facilitates a user to permit access to an application to accomplish approved functions on behalf of the user [9]. Accordingly, it empowers an external application to gain restricted access to an HTTP service. A typical OAuth use case for mobile computing and communication and its corresponding steps are illustrated in Fig. 2 [2].

C. OpenID Connect (OIDC)

OpenID Connect is a framework for transmitting identity by using RESTful APIs [7]. OpenID Connect is not a new protocol rather it is a successor of OpenID 2.0 and developed as a profile of OAuth 2.0 [7]. OpenID Connect uses two main tokens: an Access Token and an ID Token. A typical use case of OIDC for mobile computing and communication and its corresponding steps are illustrated in Fig. 3 [2].

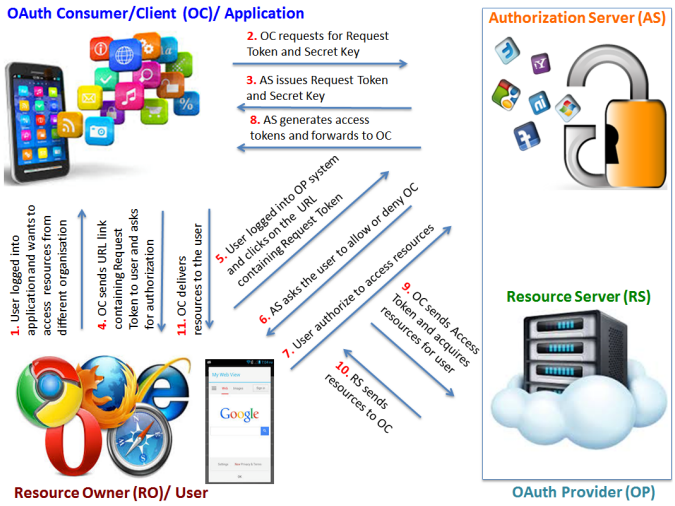

Fig. 2. An OAuth Use Case for Mobile Computing and Communication

III. Comparative Analysis of IAM Standards: SAML, OAuth and OIDC

Table I illustrates the comparative analysis of the three predominant IAM standards SAML, OAuth and OIDC on the basis of several explored criteria for this particular analysis [1], [2].

IV. Suitability Analysis of SAML, OAuth and OIDC for Mobile Computing and Communication

The previous analysis has revealed the features, merits and limitations of SAML, OAuth and ODIC standards. SAML and OIDC both are a complete solution for the authentication and authorization in for mobile computing and communication system, though OAuth should be used for an authorization, therefore, this analysis excludes it as an option.

An IAM standard should be a lightweight standard for mobile computing and communication systems. They should be able to apply data compression for reducing their size and network traffic by removing unnecessary data [6], [10]. It should be compact and faster to manage as compared to other communication protocols used in mobile communication networks. SAML is an XML-oriented specification and the representation of XML trees is quite verbose. Every element of a tree is surrounded in a pair of tags with its name/ element type. While OIDC is a JSON-oriented specification and the representation of JSON trees is less verbose than XML as it is in the form a nested array type analogous to that of JavaScript. Consequently, the more compact size of OIDC makes it the preferred choice for communication in HTML and HTTP environments than SAML [11].

TABLE I
COMPARATIVE ANALYSIS OF IDENTITY AND ACCESS MANAGEMENT STANDARDS: SAML, OAUTH AND OIDC

Criteria	SAML	OAuth	OIDC
1. Introduction Year	2005	2012	2014
2. Authentication and Authorization	It is a standard for authentication and authorization.	It is a standard for authorization (delegation) of resources.	It is a standard for authentication and authorization.
3. Main Purpose	Identity and Access Management (IAM), Single Sign-On (SSO) for enterprise users.	API authorization between Applications.	Identity and Access Management (IAM), Single Sign-On (SSO) for consumers.
4. Token Format	XML	XML, JSON, JWT	JSON, JWT
5. Token Content	Token contains user identity information but not credentials.	Token contains user identity information but not credentials.	Token contains user identity information but not credentials.
6. Security of Token (Integrity/Non-repudiation)	XML Signature HMAC or X.509; SAML tokens are almost always signed with a private key, as it is a trusted relationship between IdP and SP.	Default bearer token has no proof of possession. However, token contents can be protected by using a DS or a MAC.	JSON Web Signature (JWS)-HMAC SHA-256 or X.509; [Additional Support -RSA SHA-256 and ECDSA P-256 SHA-256].
7. Security of Token (Confidentiality/ Privacy)	XML Encryption- Triple-DES-CBC with 192-bit key and a 64-bit initialization Vector (IV), AES-CBC with a 128-bit initialization vector (IV); [TLS-SSL, Web Services Security (WSS)].	TLS is mandatory to implement with OAuth for token confidentiality. However, token encryption must be applied in addition to the usage of TLS protection.	JSON Web Encryption (JWE)-RSA-PKCS1-1.5 with 2048-bit key, AES-128-CBC, and AES-256-CBC; [Additional Support-ECDH-ES with 256-bit key, AES-128-GCM, and AES-256-GCM].
8. Lightweight Standard	It is not a lightweight standard. XML states trees in a verbose form. Every element in the tree has a name (the element type name), and the element must be enclosed in a matching pair of tags.	It is a lightweight standard. JSON states trees in a nested array type of notation similar to that of Javascript. Indeed, a JSON document can exactly be parsed as Javascript to result in the corresponding array.	It is a lightweight standard. Similar to OAuth. JSON has a much smaller grammar and maps.
9. Protocol Used	XML, HTTP, SOAP	JSON, HTTP, REST	JSON, HTTP, REST
10. Schemas and Deployments	SPML, SCIM	SCIM	SCIM
11. Platform Independent, Vendor-Neutral and Open Standard	It is a platform independent, vendor-neutral and open standard. However, flexibility in the implementation leads to the different design models.	It is a platform independent, vendor-neutral and open standard. However, flexibility in the implementation leads to the different design models.	It is a platform independent, vendor-neutral and open standard. It also standardised many parameters such as instance scopes, endpoint discovery, and dynamic registration of clients, which were left up to implementers in the OAuth 2.0 implementation.
12. Web and Native Mobile Apps Support	It is specially designed for Web apps. However, HTTP artifact binding can be used to reduce the flow of SAML messages through the browser.	It supports both Web and native mobile Apps.	It supports both Web and native mobile Apps.
13. Mobile Standard	It is limited in its ability to support mobile and smart-TV devices.	It has been designed for the mobile API and therefore it is also known as a token in your mobile.	It has been working towards standardising a GSMA Mobile Connect standard for mobile devices.
14. Enterprise and Consumer Support	It mainly supports enterprise users because it involves SP and IdP.	It supports enterprise users, and consumer apps and services.	It supports enterprise users, and consumer apps and services.
15. Fixed and Mobile Telecom Examples	British Telecom, France Telecom, Deutsche Telekom, NTT DoCoMo, National IT and Telecom Agency of Denmark.	Deutsche Telekom, Orange, T-Mobile, Tata, AT&T, Vodafone, Telecom Italia, TAT&T, France Telecom, China Mobile, China Telecom, Etisalat, KDDI, Telenor, Telefonica, Telstra.	Deutsche Telekom, Orange, Vodafone, Telecom Italia, AT&T, France Telecom, China Mobile, China Telecom, Etisalat, KDDI, Tata, Telefonica, Telenor, Telstra.

Mobile-based Single Sign-On (SSO) is one of the essential requirements for mobile users. Mobile users access a large number of apps and services and for them they require distinct authentication and authorization [12]. This leads to several issues such as remembering many passwords, regular login to the same app or service, regular password change, phishing, and password recovery. As a result, organizational productivity is affected adversely. One of the common solutions is Single Sign-On (SSO) which offers a centralized, and user-friendly and secure method of authenticating [13]. SAML and OIDC both are capable to offer the SSO functionality but OIDC is a relatively more user-friendly SSO approach for small mobile devices.

Mobile applications are a combination of applications written in native, web and hybrid languages. JSON is derived from JavaScript programming language, and its parsers are most commonly available in other programming languages, because they map directly to objects. XML was not designed for programming and the sole purpose of XML was to transport the data over on the Web. Therefore, it does not have a natural document-to-object mapping. JWT is an industry standard and is universally accepted on the Internet, offering the simplicity of client side handling of the JWT on multiple platforms, particularly, mobile [11]. Therefore, the use of JWT in the development of mobile apps is common as it is relatively more compatible than SAML assertions.

Mobile browsers are constrained in the maximum URL size they can support. Additionally, WebView has many limitations such as preventing the sharing of cookies, certificates, and HTML local storage. OIDC uses JWT that does not use **sessions**, therefore, it has no issues with native mobile applications and WebView. Thus, OIDC specification is suited for both Web browsers and WebViews (native mobile apps) whilst SAML is only for Web browsers. However, HTTP Artifact binding can be used to reduce the flow of SAML messages through the browser. Alternatively, SAML can be used for only authentication and, subsequently, OAuth can be used for authorization, where the SAML assertion can be used as the OAuth bearer token in the HTTP bearer header to access protected resources.

Mobile communication is substantially based on the use of mobile devices, where, the authentication and authorization heavily rely on mobile objects, protocols and standards. SAML offers an inadequate support for mobile devices because of its old construction as it was developed in 2005, before the introduction of first smart phone. While OIDC was introduced recently in 2014 and offered features for mobile, IoT and web. OIDC has also standardised a separate version mobile device known as GSMA Mobile Connect standard.

The authentication and authorization requirements may vary from one business model to another such as enterprise-to-enterprise, enterprise-to-consumer, or within an enterprise. A large segment of the mobile communication market is associated with consumers only; therefore, any IAM standard should provide support to consumers' authentication and authorization. The architectural design of SAML requires service provider (SP) enterprise and identity provider (IdP) enterprise, and a trustworthy relationship, therefore, it is mostly suitable for enterprise users (i.e., enterprise-to-enterprise). While, OIDC design is also focused on end users and, therefore, it is suitable for both enterprises and consumers and all business models in case of untrusted third party association.

Security is always a great concern in mobile communications due to its insecure channels and it is more disposed to eavesdropping attacks [14]. It requires two main protections; confidential information should be protected from revelation to unauthorized users, and the protection of security tokens which should not be tampered with or altered during its entire life cycle. For maintaining these two security provisions, strong encryption techniques and digital signatures or MAC should be incorporated in an IAM standard. SAML XML tokens can be signed using XML Signature (XML-Sig) based on a secret key (using the HMAC algorithm) or a public/private key pair (in the form of a X.509 Certificate). In practice, SAML tokens are generally signed with a private key because of the established relationship between IdP and SP. SAML XML token data can be encrypted using XML Encryption (XML-Enc) based on a secret key (Triple-DES-192, AES-128) or public/private key pair (RSA-PKCS1-1.5-192, RSA-OAEP-128/256). However, signing a part of the message, creating an overlapping signature and adding or subtracting text after signature features make it vulnerable for a number of new security threats. Furthermore, computing and verifying XML signatures is very resource intensive [15]. OIDC JSON Web Tokens can be signed using JSON Web Signature (JWS) based on a secret key (with HMAC algorithm) or a public/private key pair (in the form of a X.509 Certificate). OIDC JWT data can be encrypted using JSON Web Encryption (JWE) based on a secret key (AES-128-CBC, and AES-256-CBC) or public/private key pair (RSA-PKCS1-1.5-2048, ECDH-ES-256). However, some JWT libraries treat tokens signed with the **none** algorithm as a valid token with a verified signature, which allows arbitrary account access on some systems [16]. SAML and OIDC both offer strong security features. Nonetheless, complexity in signing XML with XML Digital Signatures may leave some security holes as compared to the simplicity of signing JSON [11]. Moreover, JWT does not use **sessions** while SAML does; which prevents OIDC from many attacks related to **sessions** including Cross-Site Request Forgery (CSRF), thus, more secure for mobile computing and communication.

Based on the previous two analyses, it is obvious that SAML requires a revamp in order to make it suitable for mobile computing and communication. OAuth is suitable for an authorization only but not for an authentication. OAuth is a supportive protocol for both SAML and OIDC. It is already an essential component of OIDC specification while, it can also be used with SAML to make it more suitable and lightweight for mobile computing and communication. OpenID Connect would be the most suitable choice for mobile computing and communication because it satisfies most of the requirements for them. Nonetheless, it is a developing standard and requires

wider acceptance for becoming an established IAM standard for mobile computing and communication.

V. Security Vulnerability Analysis of Identity and Access Management Standards

This analysis illustrates that flawed deployment of the powerful SAML and OIDC (OAuth is a part of OIDC) frameworks may be easily exploited for attacks.

A. Denial-of-Service (DoS) Attack

1) DoS Attack in SAML: SAML provides two common message flows, an SP-initiated and IdP-initiated, and two common SAML messages, an *Authentication Request* message sent from an SP to an IdP, and a *Response* message, containing a SAML assertion, sent from the IdP to the SP. An Authentication Request message can be sent from an SP to an IdP via the HTTP Redirect Binding, HTTP POST Binding, or HTTP Artefact Binding; and the Response message can be sent from an IdP to an SP via the HTTP POST Binding or the HTTP Artefact Binding [17], [18], [19]. Furthermore, SAML permits asymmetry in the message pair, allowing a different binding on the return message to that of the initiating message. The decision of which binding to use, is made according to the configuration settings at the SP and the IdP sides [20].

A DoS attack in SAML is possible when the SP-Initiated SSO (Redirect/POST Bindings) message flow is implemented. The user tries to access a resource on the SP, but the identity is managed by the IdP. Thus, the user is sent to the IdP to log on and the IdP delivers a SAML web SSO assertion for the user's federated identity to the SP. This exchange uses a Redirect Binding for the SP-to-IdP *AuthnRequest* message and a POST Binding for the IdP-to-SP *Response* message. Here, an attacker can target the IdP by sending abundance of requests by compromising valid users or an honest SP because the SAML request requires substantial processing overheads. The effort required for processing of each Response assertion is significantly greater than the effort required by an attacker to generate the request [21]. This could easily overwhelm the SAML IdP.

2) DoS Attack in OIDC: In OIDC *discovery* process, it is necessary to obtain OIDC IdP's configuration information. The OIDC IdP allows metadata discovery and therefore, it hosts its configuration information at the endpoint. In most of the implementations, the endpoint is accessible by any client/RP who is wishing to send registration request and thus, it is publicly open and possibly non-secure. Subsequently, OIDC client/RP sends an HTTP GET request to this metadata endpoint to obtain the configuration information of the OIDC IdP. The OIDC IdP sends a response which is a set of *Claims* about the OIDC provider's configuration, including all necessary endpoints and public key location information that can be used by client/RP for further communication with the OIDC IdP or the OAuth authorization server.

A DoS attack in OIDC is possible when the endpoint is publicly open and non-secure, and dynamic discovery process is allowed without any authentication. This vulnerability can be easily exploited for DoS attack on an OIDC IdP and flooded by countless dynamic discovery requests, which could easily overwhelm the OIDC IdP [22]. Additionally, this dynamic discovery process may be exploited for DoS attack on the client/RP, an attacker may try to spoof an OpenID IdP by publishing a discovery information that contains an issuer *Claims* using the Issuer URL of the OIDC IdP being impersonated, but with its own endpoints and signing keys. Thus, the client/RP can be flooded with information by attacker.

B. Man-In-The-Middle (MITM) Attack

1) MITM Attack in SAML: A MITM attack is possible in SAML when the SP-Initiated SSO (POST/Artefact Bindings) message flow is implemented. This exchange uses a POST Binding for the SP-to-IdP *AuthnRequest* message and an Artefact Binding for the IdP-to-SP *Response* message [20]. The user tries to access a resource on the SP but the identity is managed by the IdP. The user enters correct credentials and a local logon related security setting is generated for the user at the IdP. Later, the IdP creates an artefact containing the source ID for its website and a reference to the *Response* message (the MessageHandle). The HTTP Artefact binding permits the choice of either HTTP redirection or an HTML form POST as a way to deliver the artefact to the SP [17].

The SP's Assertion Consumer Service sends a SAML *ArtifactResolve* message, which contains the artefact to the IdP's Artefact Resolution Service endpoint using the synchronous SOAP binding. The IdP's Artefact Resolution Service extracts the MessageHandle from the artefact and finds the original SAML *Response* message accompanying with it [17]. The retrieved message is placed in a SAML *ArtifactResponse* message that is returned to the SP using the synchronous SOAP binding. The SP extracts and processes the *Response* message and the embedded assertion for creating a local logon security setting for the user at the SP [17].

In this SAML SP-Initiated SSO (POST/Artefact Bindings) process, the SOAP binding is used which is vulnerable to the MITM attack [23]. The *RelayState* token is not a transparent reference to state information which is maintained at the SP. This *RelayState* mechanism can leak information about the user's activities at the SP to the IdP if the SP deployment is erroneous or some other kind of existing vulnerabilities which may also lead to the MITM attack [24]. Since the HTTP Artefact binding will be used to deliver the SAML *Response* message, it is not compulsory that this assertion be digitally signed which is also a great security risk and increases the chances of the MITM attack in SAML.

2) MITM Attack in OIDC: A MITM attack is possible in OIDC when the OIDC *dynamic client registration* process is happened. For registering a new OIDC client/RP at the Authorization Server, the client/RP sends an HTTP POST message including its metadata to the Client Registration Endpoint with a content type of application/JSON, and the parameters represented as top-level elements of the root JSON object. The subsequent response may carry a Registration Access Token, which can be used by the client/RP to accom-

plish required tasks upon the resulting client/RP registration. The OIDC IdP may require an Initial Access Token to limit registration requests to only authorized clients or developers [22]. However, to support an open dynamic registration, the Client Registration Endpoint should accept registration requests without OAuth 2.0 Access Tokens. Therefore, the dynamic client registration could be the potential source of many attacks including the MITM attack.

This MITM attack may be caused by a logical flaw in the OAuth 2.0 protocol or the presence of a malicious OIDC IdP or malicious client/RP [25], [26]. A malicious OIDC IdP can trick the client/RP into sending an authorization code to the attacker's Token Endpoint. Once a code is stolen, an attacker can modify information of authorization requests and responses for confusing the RP into binding an authorization to the wrong user [27]. Consequently, the confused RP may select wrong IdP at the start of the authentication or authorization process [25], [26]. This permits a hacker to modify user data and fool the RP into treating it as the IdP the user wants [25], [26]. Accordingly, the RP sends the authorisation code or the access token issued by the honest IdP to the attacker depending on the OAuth mode employed. Finally, an attacker can utilise this information for login into the client/RP under the user's identity (managed by the honest IdP) or accessing the user's protected resources at the honest IdP [25], [26].

VI. Conclusion

This paper presented the evaluation of three popular IAM standards, SAML, OAuth and OIDC to ascertain suitable IAM to protect mobile computing and communication. This evaluation is based on the three types of analyses: comparative analysis, suitability analysis and security vulnerability analysis of SAML, OAuth and OIDC. SAML was developed before smart mobile phones were introduced, and therefore it has many legacy features, which are not compatible with mobile computing and communication and would require a revamp to make it a more suitable IAM. OAuth is suitable for an authorization only but not for an authentication. OpenID Connect would be the best choice for mobile computing and communication as it fulfils maximum requirements for them. Nonetheless, it is a developing standard and requires wider acceptance for becoming an established IAM standard for mobile computing and communication. In the future, it may be interesting to perform practical investigations on SAML, OAuth and OIDC for mobile computing and communication systems.

References

[1] N. Naik and P. Jenkins, "A secure mobile cloud identity: Criteria for effective identity and access management standards," in *2016 4th IEEE International Conference on Mobile Cloud Computing, Services, and Engineering (MobileCloud 2016)*. IEEE, 2016.

[2] ——, "An analysis of open standard identity protocols in cloud computing security paradigm," in *14th IEEE International Conference on Dependable, Autonomic and Secure Computing (DASC 2016)*. IEEE, 2016.

[3] N. Naik, "Connecting Google cloud system with organizational systems for effortless data analysis by anyone, anytime, anywhere," in *IEEE International Symposium on Systems Engineering (ISSE)*. IEEE, 2016.

[4] D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing," *SIAM Journal on Computing*, vol. 32, no. 3, pp. 586–615, 2003.

[5] I. Stojmenovic, *Handbook of wireless networks and mobile computing*. John Wiley & Sons, 2003, vol. 27.

[6] M. R. Momeni, "A lightweight authentication scheme for mobile cloud computing," *International Journal of Computer Science and Business Informatics*, vol. 14, no. 2, 2014.

[7] Pingidentity.com. (2011) A standards-based mobile application idm architecture. [Online]. Available: http://www.enterprisemanagement360.com/wp-content/files_mf/white_paper/exp_final_wp_mobile-application-idm-arch-8-11-v4.pdf

[8] N. Klingenstein, T. Hardjono, H. Lockhart, and S. Cantor. (2012) OASIS Security Services (SAML) TC. [Online]. Available: https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=security

[9] N. Ranjbar and M. Abdinejadi, "Authentication and Authorization for Mobile Devices," B.Sc. Dissertation, Department of Computer Science and and Engineering Goteborg, Sweden, 2012.

[10] N. Naik, "Migrating from virtualization to dockerization in the cloud: Simulation and evaluation of distributed systems," in *IEEE 10th International Symposium on the Maintenance and Evolution of Service-Oriented and Cloud-Based Environments, (MESOCA)*. IEEE, 2016, pp. 1–8.

[11] JWT.io. (2015) Introduction to JSON web tokens. [Online]. Available: https://jwt.io/introduction/

[12] N. Naik, "Building a virtual system of systems using Docker Swarm in multiple clouds," in *IEEE International Symposium on Systems Engineering (ISSE 2016)*. IEEE, 2016.

[13] A. Gunawardana. (2014, July 18) SSO for native mobile applications with WSO2 identity server. [Online]. Available: http://wso2.com/library/articles/2014/07/sso-for-native-mobile-applications-with-wso2-identity-server/?gclid=CKje4-3P1skCFdRuGwod2qYGhw

[14] N. Alsulami and M. M. Monowar, "Data confidentiality and integrity in mobile cloud computing," *Journal of Emerging Trends in Computing and Information Sciences*, vol. 6, no. 3, pp. 138–143, 2015.

[15] W3.org. (2015) XML security ? issues and requirements. [Online]. Available: http://www.w3.org/2007/xmlsec/ws/papers/09-lockhart-bea/

[16] T. McLean. (2015) Critical vulnerabilities in JSON web token libraries. [Online]. Available: https://auth0.com/blog/2015/03/31/critical-vulnerabilities-in-json-web-token-libraries/

[17] Oasis-open.org. (2008, March 25) Security Assertion Markup Language (SAML) v2.0 technical Overview. [Online]. Available: http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-tech-overview-2.0.html

[18] N. Naik, P. Jenkins, P. Davies, and D. Newell, "Native web communication protocols and their effects on the performance of web services and systems," in *Computer and Information Technology (CIT), 2016 IEEE International Conference on*. IEEE, 2016, pp. 219–225.

[19] N. Naik and P. Jenkins, "Web protocols and challenges of web latency in the web of things," in *Ubiquitous and Future Networks (ICUFN), 2016 Eighth International Conference on*. IEEE, 2016, pp. 845–850.

[20] J. Somorovsky, A. Mayer, J. Schwenk, M. Kampmann, and M. Jensen, "On breaking saml: Be whoever you want to be." in *USENIX Security Symposium*, 2012, pp. 397–412.

[21] J. Naithan, "SAML proposal for securing XML web services.Project paper," *University of St. Thomas, Saint Paul*, 2008.

[22] V. Mladenov, C. Mainka, and J. Schwenk, "On the security of modern single sign-on protocols: Second-order vulnerabilities in openid connect," *arXiv preprint arXiv:1508.04324*, 2015.

[23] T. Groß, "Security analysis of the saml single sign-on browser/artifact profile," in *Computer Security Applications Conference, 2003. Proceedings. 19th Annual*. IEEE, 2003, pp. 298–307.

[24] SANS. (2003) Global information assurance certification paper. [Online]. Available: https://www.giac.org/paper/gsec/2876/saml-common-security-language-web-services/104846

[25] D. Fett, R. Küsters, and G. Schmitz, "A comprehensive formal security analysis of oauth 2.0," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2016, pp. 1204–1215.

[26] R. Millman. (2016, January 11) Researchers find two flaws in OAuth 2.0. [Online]. Available: https://www.scmagazine.com/researchers-find-two-flaws-in-oauth-20/article/530038/

[27] G. Curtis. (2016, July 16) Preventing mix-up attacks with OpenID Connect. [Online]. Available: http://openid.net/2016/07/16/preventing-mix-up-attacks-with-openid-connect/