A Centralized Multimodal Unified Authentication Platform for Web-based Application

Sea Chong Seak, Ng Kang Siong, Wong Hon Loon, Galoh Rashidah Haron

***Abstract*— Identifying a user or in technical terms, authentication, is a key process that must be performed before a service can be rendered to a person through electronic means. Common practice for user authentication is to challenge user to present authentication credential, user who can present the valid credential is considered as authenticated identity. Traditionally, each application handles the authentication process by itself. However, recent development in information technology manages to decouple authentication function from application, thus allowing for new possibilities for the application to counter the internet fraud in a more cost effective way. The organization Information System Security research lab has developed Unified Authentication Platform (UAP) based on Security Assertion Markup Language (SAML) 2.0 specification that support multimodal authentication mechanisms with single sign-on and adaptive control based on security risk and level of assurance.**

***Index Terms*— User Identification, multimodal authentication, Adaptive authentication, SAML, Unified Authentication Platform, Web authentication, Single Sign-on**

I. Introduction

USER authentication is common basic requirement for modern web-based applications as more and more personalized and access controlled services to move it online. Cloud applications have become popular among organizations who share content on the Internet. Most of the organizations have started using a central authentication source for their web portal and web applications. The single source of centralized authentication when configured properly, it can provides strong security protection in the sense that users no long keep user credential such as passwords for different application systems on yellow sticky notes put on the LCD monitor or hide under their keyboards. In additional benefits, users management and auditing becomes more simplifies to manage.

One of our goals is to provide application architectures and implementers with the clear framework within which to think about and develop secure web based authentication systems. This due to most of the web applications developed their own authentication system to provide a better user experience. However, the web application architectures and implementers usually do not have a solid background especially in security areas; as a result do not have a good understanding of the tools at their disposal.

Between organization web applications and centralized authentication system, it is apparent that a solution is required to provide a standard for authentication information to be exchanged over the internet. Security Assertion Markup Language (SAML) provides a secure, XML based solution for exchange user security information between authentication system and applications. In general, SAML standard defines rules and syntax for data information exchange, yet is flexible and allow for custom data to be transmitted to applications. Using SAML, an online web application can contact separate authentication provider to authenticate users who are trying to access secure content.

II. Background

A. *Security Assertion Markup Language (SAML)*

The consortium defining SAML standards and security is OASIS (Organization for the Advancement of Structured Information Standard) https://www.oasis-open.org/standards. They are a non-profit international organization that promotes the development and adoption of open standards for security and web services.

SAML version 2.0 [1] was approved as an OASIS Standard in March 2005. SAML is an XML-based open standard data format for exchanging authentication data between parties. The main functions of SAML 2.0 address is to enabled web-based authentication scenarios including cross domain web browser Single Sign-On (SSO), which assist reduce the administrative overhead of distributing multiple authentication security token to user.

B. *Shibboleth*

The Shibboleth project [2] began as an Internet2 Middleware activity in 2000, and later that year the project

Manuscript received July 21, 2014; revised August 08, 2014.

Sea Chong Seak is with Information Security Lab, MIMOS Berhad, Technology Park Malaysia, 57000 Kuala Lumpur, Malaysia. (e-mail: cssea@mimos.my).

Ng Kang Siong is with Information Security Lab, MIMOS Berhad, Technology Park Malaysia, 57000 Kuala Lumpur, Malaysia. (e-mail: ksng@mimos.my).

Wong Hon Loon is with Information Security Lab, MIMOS Berhad, Technology Park Malaysia, 57000 Kuala Lumpur, Malaysia. (e-mail: hl.wong@mimos.my).

Galoh Rashidah Haron is with Information Security Lab, MIMOS Berhad, Technology Park Malaysia, 57000 Kuala Lumpur, Malaysia. (e-mail: rashidah@mimos.my).

connected with the work of the OASIS SAML Working Group.

Shibboleth is a free open-source project released under the Apache Software License that provides web browser Single Sign-On capabilities and allows sites to make informed authorization decisions for individual access of protected online resources in a privacy-preserving manner. Shibboleth popular feature attribute-exchange based on open standards, principally Security Assertion Markup Language (SAML), it is a federated system, supporting secure access to resources across security domains. The Shibboleth architecture defines a way of exchanging information between an individual and a provider of digital data resources. Shibboleth can protect both security to the data and privacy of the individual viewing it.

Diagram at Figure 1 shows the flows which can take place during a typical Shibboleth-enabled transaction [3]. Begin with the browser, user arriving at the Service Provider (SP) site without an existing session and without any information about the user's home institution being known by the Service Provider. Information about a user is sent from an identity provider (IdP) to a SP which prepares the information for protection of sensitive content and use by applications.

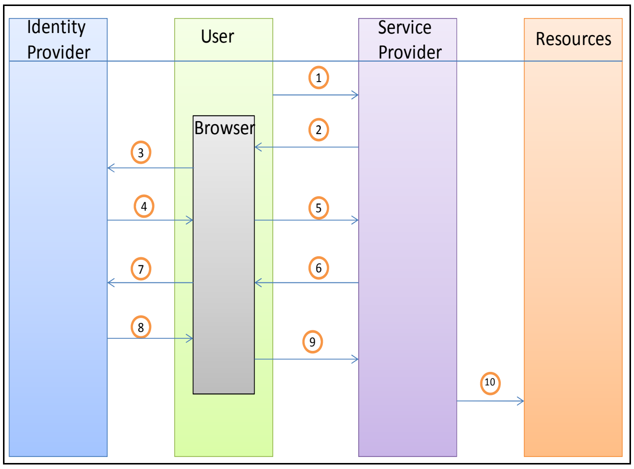

Figure 1: Shibboleth-enabled transaction Flows Diagram

1. The user attempts to access a Shibboleth-protected resource on the Service Provider.
2. Service Provider generates SAML request and redirects browser to Identity Provider URL.
3. Browser is redirected to Identity Provider.
4. Identity Provider parses SAML request and then by whatever means it deems appropriate, ensures that the user is authenticated. After successful authentication, a one-time handle (session identifier) is generated for this particular user and is send to Service Provider via browser.
5. Browser send handle (session identifier) to Service Provider).
6. Service Provider uses handle request user attributes information from the Identity Provider.
7. Browser sends user attributes information to Identity Provider.
8. Identity Provider, based on its Attribute Release Policy, allows or denies attribute information to be made available to this Service Provider.
9. Browser sends user attributes information to Service Provider.
10. Service Provider allows or refuses user access to the resources based on user attributes information.

C. *Authentication framework*

There are numerous of authentication framework and single Sign-on technologies widely used in web-based applications. However, the authors have categories it into three major frameworks as listed below:

Authentication via common databases

User gets authenticated to application A and B by using the same password. Both applications compare and validate user input credential via the same common password storage.

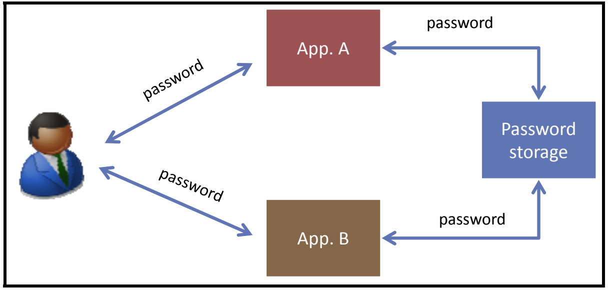

Figure 2: Authentication via common databases

Authentication via common gateway

This framework user requires a password to login to the Single Sign-on gateway. But user also must deposit and store the password credential for application A and B in gateway password storage. During the authentication, gateway will post the password individually to the application for user authentication. The entire user password must store in plain text in gateway databases. All the applications must have individual password storage to store user credential.

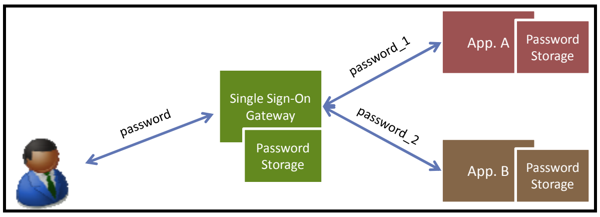

Figure 3: Authentication via common gateway

Authentication via SAML

All the allocation will protected by authentication gateway, authentication and validation user credential happened in authentication server. SAML is apparent that a solution is required to provide a standard for authentication information to be exchanged between the gateway and authentication server. Persistence ID generate by authentication server and then will use by the application to perform authorization.

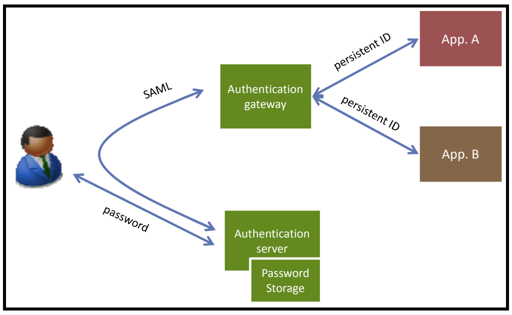

Figure 4: Authentication via SAML

D. *Features comparison of Authentication frameworks*

The authors of this paper have done some features comparison between the three authentication frameworks. Table 1 below indicates the summary of features comparison between three authentication frameworks.

Table 1: Features comparison of authentication frameworks

Features	Common Database	Common Gateway	SAML
Single sign-on	Yes	Yes	Yes
Password privacy protection	Low (Visible to application)	Low (Visible to application)	High (Not visible to application)
Direct interaction with user	No	Yes	Yes
Require plaintext password storage	No	Yes	No
Password synchronization	Not required	Required	Not required
Additional authentication module	Extensive changes required at each application	No Change required at each application	No Change required at each application
Integration	Modification of application required	No modification of application required	Modification of application required

III. UNIFIED AUTHENTICATION PLATFORM

The organization Information System Security research lab has developed a centralized multimodal Unified Authentication Platform (UAP) based on Security Assertion Markup Language (SAML) 2.0 specification [1]. UAP addresses problems related to the increase of operational risks attributed to users and system administrators who control and provide cross-application functionalities in heterogeneous applications. UAP is designed to manage front-end application authentication using SAML protocol which provides a centralized authentication framework and aims to reduce significant application changes at the back-end. Technology Benefits UAP simplifies the complexity of managing multiple user IDs with seamless identity management using a trust model approach. UAP also utilizes Single Sign-On (SSO) for a seamless environment operation and adaptive authentication and threat response without modifying existing applications.

UAP has equivalent functionality as the Shibboleth [2] Identity Provider (IDP) [3], which is responsible for user authentication and providing user information to the Shibboleth Service Provider (SP). It is also responsible to maintain the user's account, for privacy reason UAP only require minimum user information just enough for identity verification.

A. *Objective of UAP*

The objectives of doing research and development project on a centralized multimodal unified authentication platform are:

1) Provide an infrastructure to provide authentication service to applications.
2) Provide authentication mechanism de-couples from application.
3) Grow indigenous authentication mechanism industry throughout the country.
4) A unified authentication platform initiative for enabling government e-services application.
5) Improved privacy compliance by allowing the user to control what information is shared, or by limiting the amount of information shared.

B. *Problem Statement*

During our initial research finding, we have found some challenges and problems associated with current authentication scenario.

The major problems are:

1) Each application handles user authentication in silo.
2) Huge migration cost to support additional authentication mechanisms.
3) Lack of expertise to keep up-to-date security threat on specific user authentication mechanisms.
4) Lack of common platform to grow indigenous identity and authentication security industry to enhance national competitiveness.

IV. UAP ARCHITECTURE

UAP provides an infrastructure to provide authentication services to applications with the authentication mechanism decoupled from application implementation. This is a self-managed system which incurs zero administrative effort. It allows users to self-manage user accounts and the authentication methods associated with the account.

UAP system architecture comprises four major components:

1) Authentication Server executes authentication interactions based on a user's presentation of credentials, and subsequently computes the PID to be used by the Authentication Gateway server and the protected application.
2) Authentication Gateway accepts incoming requests on behalf of a protected application and checks the authenticated session for a particular user.
3) Application Server executes and handles all application operations between users and an organization's backend business applications or databases.

4) User accesses the web application and authentication services via web browser. In the Mi-UAP system, a user may require additional hardware components i.e. smart phone for mobile application (TCK and 2Dbarcode), mobile phone for OTPs delivered via SMS, token for OTP generation, digital certificate (relating to cryptographic public-key) as might be contained in a PKCS-11 compliant smartcard i.e. the national registration identity card (MyKAD) [12].

Diagram at Figure 5 shows high level of the UAP system architecture. UAP introduces six authentication methods available for users to choose. The authentication method implemented in UAP depicted in Figure 5 includes four common authentication methods used today which is password, certificate, One Time Password (OTP) token and OTP SMS; and other two proprietary authentication methods was introduce and developed by the authors which is Time-Constrained Key (TCK) and 2DBarcode.

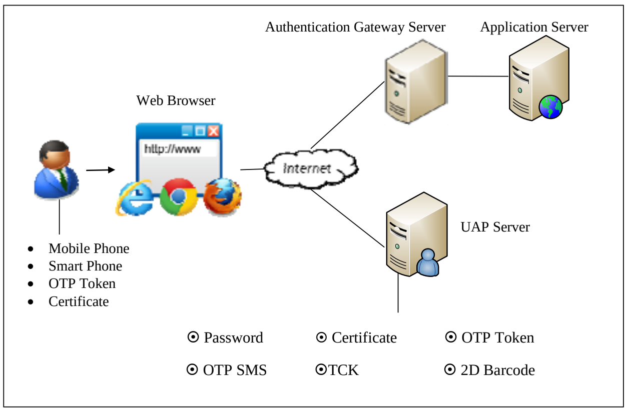

Figure 5: UAP System Architecture

A. *Password*

The most basic authentication method relied on the user chooses it and something that fits in the memory of a user. This method of authentication is about verifying the user physical identity remotely, and the user behavior is necessarily involved throughout the process. "Password strength" can be somewhat improved by mandatory rules (E.g. a password must has minimum 8 character with combination upper and lower case with at least one numeric number). This is a cheapest and easy way to implement which no longer require any hardware token.

B. *Certificate*

At the very least, a certificate is a signed data structure that contains someone`s name, public key and asserts these entities are bond together. Additional, certificate also contains a validity interval that status how long the certificate is valid for.

The certificate is issued by a license Certification Authority (such as VeriSign, GlobalSign or MSC Trustgate), who guarantees the link between a physical identity and a cryptographic public key. The verifier may be a distinct entity, and can verify such a link and use it to authenticate the user, without getting the ability to impersonate the user.

A cryptographic smart card that is capable of performing RSA private key operations using the stored private key. Malaysian national identity card, MyKAD, is capable of performing such operation and therefore can also be used for the purpose stated in this paper. The smart card can also be replaced by virtual memory storage of private key with the necessary cryptographic functions to perform the similar private key operations.

A program developed by the authors of this paper, PKCS#11/CSP – PKCS#11[14] is a cryptographic token interface library that can be loaded into Mozilla Firefox while CSP [15] serves the same purpose for Microsoft Internet Explorer. These cryptographic token interface libraries allow the browsers and browser extension programs to interact with cryptographic tokens to perform RSA private key related operations that involved the use of smart card or virtual memory storage.

The benefits of having this authentication method are after the certificate is configured, there is nothing further to be done. When a user tries to log on or access a gated application/network, he will be prompted to select his certificate from a list. Certificate easily enable two-factor authentication across multiple applications and network and support mobile/remote workforce.

C. *One Time Password tokens*

A password that is valid for only one login session or per single transaction. OTP numbers are difficult for human beings to memorize. Therefore, they require additional technology and hardware token in order to work.

The OTP device creates a "two-factor" security system which means you’ll have to know something (your user name, password and PIN) and have something (the OTP security token) in order to login into the authentication.

D. *One Time Password SMS (Mobile SMS OTP)*

The method used to authenticate user based on the non-reusable random generated mobile Short Message System (SMS) OTP deliver to a user via SMS. Mobile SMS OTP will only be valid per login session. After successfully login, the authentication server will generate brand new mobile SMS OTP and send it to a user via mobile SMS network.

This method also creates a "two-factor" security system, which the mobile SMS OTP and the user mobile phone to receive the mobile OTP via the SMS network.

E. *Time-Constrained Key (TCK)*

New method introduced and developed by the authors to perform user authentication. TCK will feature as OTP (one time password) generation and verification implemented by means of only symmetric cryptographic mechanisms.

The overview TCK system architecture is depicted in Figure 6. TCK is a time-based One Time Password (OTP) generator application for the mobile device.

This solution is based on time synchronous one time passwords. When click on TCK client application installed in user mobile device, the mobile application generate 8 character of alpha numeric OTP code and display on the screen. The OTP code can be verified by the TCK server, as the server knows the current time and user secret key. To compensate time differences, the TCK server will accept OTP code from 2 minutes in the past to 2 minutes in the future. In addition, different time offsets can be specified for each user on the client and/or the server. Each password will be accepted only once.

TCK mobile application can have a pin protected, user must enter a valid pin each time he/she click to generate OTP code. After 5 successive failed authentication attempts a user gets locked out. This authentication method is based on two factors which is a pin know by user and the secret key stored on the mobile device.

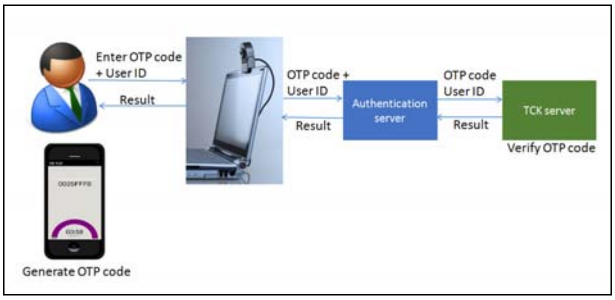

Figure 6: TCK system architecture diagram

OTP code generation and verification should also be undertaken with system time representations in UTC or GMT, so as to allow for authentication independent of user location as approximated by the client-side time zone configuration at instance of authentication.

TCK client application (Mi-TCK) current available download from Apple IOS App Store, (https://itunes.apple.com/hk/app/mi-tck/id875191492?mt=8) and Android Play store (https://play.google.com/store/apps/details?id=my.mimos.isluap.mitck).

The benefits use TCK are no additional hardware require can use existing smart phone (iOS and Android) and no extra cost for each OTP code generated.

F. *2DBarcode*

Another new method introduced and developed by the authors to perform user authentication. This is a novel method of advanced authentication method use of two-dimensional barcode as a security token to perform user authentication via visual channel. The communication and transaction of data perform via a visual channel. This type of authentication also called “Out-of-Band” (OOB) authentication, which is OOB uses a completely separate channel, such as smart phone to authentication a transaction originated from a computer.

The overview 2DBarcode system architecture is depicted in Figure 7. 2DBarcode is a time-based One Time Password (OTP) generator application for the mobile device. This solution also based on time synchronous one time passwords. When click on 2DBarcode client application installed in mobile devices, the mobile application generate 16 byte of OTP code then append together with user ID encoded in 2Dbarcode format and display on the screen.

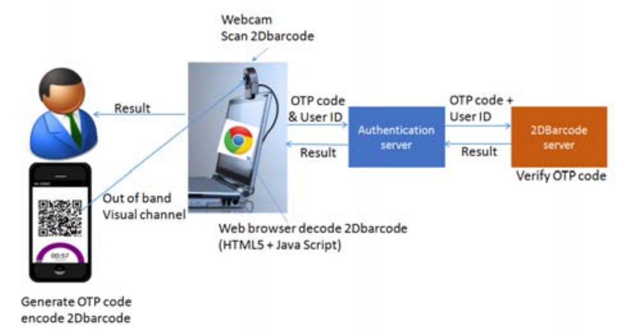

Figure 7: 2DBarcode system architecture diagram

To compensate time differences, the 2DBarcode server also will accept OTP code from 2 minutes in the past to 2 minutes in the future. 2DBarcode generation and verification should also be undertaken with system time representations in UTC or GMT, so as to allow for authentication independent of user location as approximated by the client-side time zone configuration at instance of authentication.

2DBarcode would implement the following transport mechanisms, visual scan: with

- Representation of barcode for display on device screen, and subsequent
- Capture (via camera) and decode (via browser HTML5 & java script).

2DBarcode client application named (Mi-2DBC) current available download from Apple IOS App Store, (https://itunes.apple.com/us/app/mi-2dbc/id875650852?mt=8) and Android Play store (https://play.google.com/store/apps/details?id=my.mimos.isluap.mi2dbc&hl=en).

V. Integration and Development Experiences

Diagram in Figure 8 demonstrates how in house enterprise applications integrated with UAP to provide multiple factor authentication and single sign-on services. User has option to activate all six authentication methods and also has option to use any method for authentication.

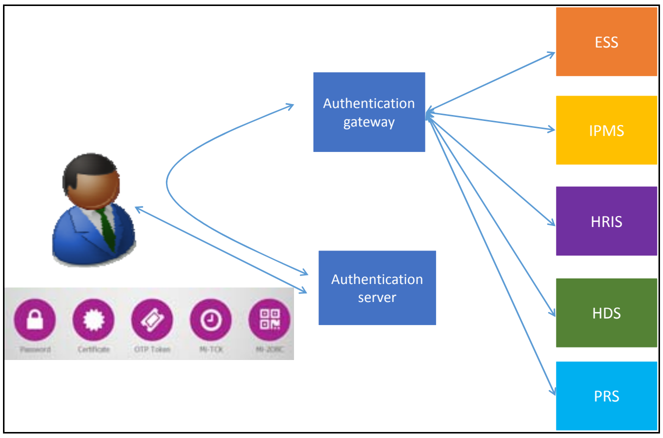

Figure 8: Integration organization enterprise application with UAP

The experience we gain from this integration and deployment projects was able to turn multiple platform applications to use UAP for authentication and single sign-on. The applications already integrated with UAP as listed in Figure 8 are Employee Self Service (ESS) application is SAP based application, Intellectual Property Management System (IPMS) and Human Resource Information System (HRIS) was developed using Microsoft .Net platform and Helpdesk System (HDS) and Purchase Requisition System (PRS) was developed using JAVA programming.

In this integration tasks, UAP provide an infrastructure to organization enterprise applications developed based on different platforms and programming languages using UAP authentication services. Finally, UAP successfully demonstrate authentication mechanism de-couples from application and provide application subscribe to UAP authentication services.

VI. Conclusion

Today's enterprises require a choice, and are looking for versatile solutions that meet their needs for strong security, usability, cost control and ease of integration.

The increase in cybercrime necessitates an increase in security measures. Fortunately, the above authentication methods make it much harder for a criminal gang to exploit their targets, which will hopefully save millions per year in lost revenue and productivity.

Both TCK & 2DBarcode are time-based One Time Password (OTP) generator application for the mobile device. It is the client mobile application highly secure, simple to use and administer, and extremely cost effective solution for meeting your strong authentication needs. It makes the end-user mobile device behave like a hardware based one-time-password OTP token without the hassle of carrying one, while providing easy administration and high security for web authentication.

The similar architecture and concept can deploy in larger organization or government agency for flexibility selection of authentication methods with security protection.

References

[1] N. Ragouzis et al., Security Assertion Markup Language (SAML) V2.0 Technical Overview. OASIS Committee Draft, March 2008. Document ID sstc-saml-tech-overview-2.0-cd-02 <http://www.oasis-open.org/committees/download.php/27819/sstc-saml-tech-overview-2.0-cd-02.pdf>

[2] Shibboleth Consortium, “Shibboleth”, <http://shibboleth.net>

[3] Shibboleth Consortium - Identity Provider , <https://shibboleth.net/products/identity-provider.html/>

[4] OASIS, “OASIS Security Services (SAML) TC” , https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=security

[5] Gross, T, “Security analysis of the SAML single sign-on browser/artifact profile”, pp. 298 – 307, Computer Security Applications Conference, 2003. , pp. 298–307

[6] Harding P. , Johansson L. , Klingenstein N., “Dynamic Security Assertion Markup Language: Simplifying Single Sign-On”, IEEE Security & Privacy March – April 2008, vol 6 issues 2, pp 83-85.

[7] Fugkeaw. S , Manpanpanich P. , Juntapremjitt S., “Adding SAML to two-factor authentication and single sign-on model for dynamic access control”, 6th International Conference on Information, Communications & Signal Processing, 2007, 10-13 Dec. 2007, pp. 1–5.

[8] Wu Kaixing,Yu Xiaolin, “A Model of Unite-Authentication Single Sign-On Based on SAML Underlying Web” Second International Conference on Information and Computing Science, 2009. ICIC '09. 21-22 May 2009. Vol 2 , pp.21 1–213.

[9] Wang Jun, Del Vecchio D. , Humphrey M. , “Extending the security assertion markup language to support delegation for Web services and grid services,” IEEE International Conference on Web Services, 2005. ICWS 2005. 11-15 July 2005, Vol 1 , pp.67–74.

[10] Zhenxiang Tu, Qian Li, “Design and implementation of unified identity management system based on SAML,” 2nd International Conference on Consumer Electronics, Communications and Networks (CECNet), 2012, 21-23 April 2012, pp.3178–3181

[11] Alessandro Armando,Roberto Carbone, Luca Compagna, Jorge Cuellar,Llanos Tobarra ,“Formal Analysis of SAML 2.0 Web Browser Single Sign-On: Breaking the SAML-Based Single Sign-on for Google Apps”, Proceedings of the 6th ACM workshop on Formal methods in security engineering 2008, pp 1-10.

[12] Mykad, “Introduction ti MyKad”, < http://www.jpn.gov.my/en/informasi/introduction-mykad>

[13] Bruce Schneier, “Applied Cryptographic”, Second Edition, John Wiley & Sons, 1996, ISBN 0-471-11709-9.

[14] RSA Laboratories, “PKCS#11 base Functionality v2.30: Cryptoki-Draft 4”, 2009. < ftp://ftp.rsasecurity.com/pub/pkcs/pkcs-11/v2-30/pkcs-11v2-30b-d6.pdf>

[15] Microsoft TechNet, “Microsoft CryptoAPI and Cryptographic Service Providers”, < http://technet.microsoft.com/en-us/library/cc962093.aspx>

[16] Jake Wu, Panos Periorellis, ‘Authorization-Authentication using XACML and SAML”, Technical Report Series, University of Newcastle, school of computing science, May 2005.

[17] Kelly D. Lewis, James E.Lewis, “Web Single Sign-On Authentication using SAML”, IJCSI International Journal of Computer Science Issues, Vol 2, 2009.

[18] Juraj Somorovsky, Andreas Mayer, Jorg Schwenk, Marco Kampmann and Meiko Jensen, “On breaking SAML: be whoever you want to be”, Security 2012 proceeding of 21st USENIX conference on Security symposium, 2012.