Tuijin Jishu/Journal of Propulsion Technology
ISSN: 1001-4055
Vol. 44 No. 4 (2023)

Openid Authentication Model for Cloud Security Using Trusted Platform

[1*]Mr. Suresh S, [2]Dr. Manish Varshney
[1]Ph.D. Research Scholar
Department of Computer Science, Maharishi School of Engg. & Tech., MUIT University, Lucknow, U.P.
E-Mail: sureshsalendra@gmail.com,
[2]Professor, Department of Computer Science, Maharishi School of Engg. & Tech., MUIT University, Lucknow, U.P.

Abstract:

Online services on the Internet have expanded incredibly quickly in recent years. To use any online service, Internet users must first register for a new account. The issue is evident when one user often requires many services and as a result maintains various accounts. To prevent identity theft, these various accounts must be managed in a safe and convenient manner. SSO (single sign-on) and OpenID have been used to make maintaining the many accounts needed in the Internet identity context less difficult. Two excellent trusted computing-based technologies to address security issues in the Internet identity context are Trusted Platform Module (TPM) and Trust Multitenancy.

Keywords: OpenID, Cloud, SSO, TPM, Trust, Authentication.

A variety of identity management solutions, including OAuth 2.0, Shibboleth, CardSpace, and OpenID, have been proposed [1–3] in an effort to lessen the harm caused by identity assaults and streamline the administration of identities. OpenID Connect 1.0 [4] adds an identity layer on top of the OAuth 2.0 framework [2] to replace the well-known OpenID [3] system. The OAuth 2.0 framework gives an RP the ability to get pro le information about the end user, but it does not provide the RP any way to acquire information about the end user's authentication. In OpenID Connect, RPs can obtain assurances about the end user's identity from an OpenID Provider (OP), which also authenticates the user, in addition to profile information about the end user.

Four key parties engage with one another in OpenID Connect:

1. The End User (U), who utilises the RP's online services;

2. The User Agent (UA), typically a web browser, used by the end user to send requests to and receive responses from web servers;

3. The OpenID Provider (OP), such as Google, which offers techniques for end user authentication and generates assertions about the authentication event and the end user's attributes;

4. The Relying Party (RP), for instance, Wikihow, which offers secured web services and uses an identity statement produced by the OP to determine whether it's safe to give access to the ultimate user.

The process of OpenID connection process as follows. OpenID Connect introduces a new kind of token to OAuth 2.0 called the id token in order to allow an RP to confirm the identity of an end user. The access

token and code from OAuth 2.0 are supplemented by this. These three different token kinds have the following purposes and are all issued by an OP.

An identification and a URL of the RP are connected to an opaque value known as a code. Giving an RP permission to get more tokens from the OP is its primary function in OpenID Connect. It has a short validity term and is often designed to expire soon after being issued to the RP in order to reduce dangers stemming from its potential exposure [2].

A credential known as an access token is used to allow access to secured materials kept by a third party, such as the OP. Its value is an opacity string that represents a permission given to the RP. It specifies the extent and length of the RP's permission to access data stored by a particular third party, as given by the end user and upheld by the RP and the OP.

An id token includes any further claims that the RP may have sought in addition to the claims about the authentication of an end user by an OP. The identification of the OP that issued it, the user's unique identity at this OP, the identity of the intended receiver, the moment at which it was issued, and its expiration period are among the claims that may be included to such a token. It is digitally signed by the OP and is presented as a JSON Web Token [5].

A call to the issuing OP's web API may be used to verify both an access token [6] and an id token [7].

On top of user agent HTTP redirections, OpenID Connect is built. Assume that a user wants to access RP services, which employ tokens produced by OP. In place of the end user, the RP creates an authorisation request and delivers it to the OP through the UA (usually a web browser). The OP offers methods for authenticating the end user, requests permission from the end user for the RP to access user attributes, and creates an authorization response with two sorts of tokens: access tokens and id tokens, the latter of which contains assertions about user authentication. After obtaining an id token, the RP learns about user authentication, as described in Fig. 1. The RP may utilise an access token obtained to access end user characteristics via the OP-provided API.

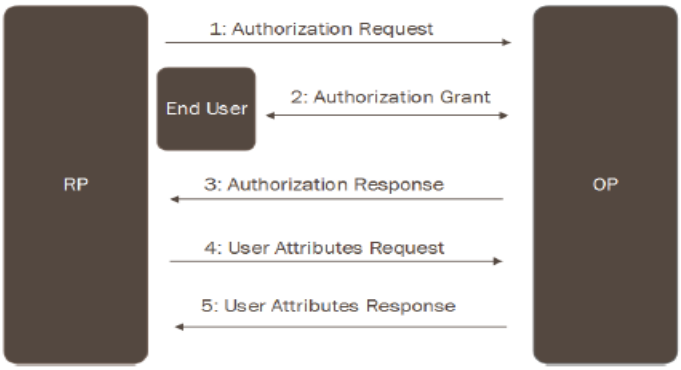

Fig. 1. OpenID Connect Protocol Overview

Using OpenID, as mentioned above, might lead to certain security issues as well. Here are a few of these security issues.

Attacks that eavesdrop: This protocol is vulnerable to eavesdropping attacks. In other words, if a nonce is not being verified, an eavesdropper may intercept a valid authentication claim and reuse it.

Denial of Service Attack: Another issue with OpenID is that because to flaws in the OpenID protocol, a malicious relying party is possible to execute a DOS attack against the OpenID provider. Since no such information is included in the messages of the OpenID protocol, the OP cannot quickly determine whether a request is genuine or not. This circumstance could arise as a consequence of the relying party repeatedly requesting authentication, associations, or signature verification.

Man in the Middle (MITM) Attack: In general, associations can stop MITM Attacks from modifying signed fields, but there are several exceptions that can place during discovery, association sessions, and direct verification. An attacker with access to a compromised DNS will be able to act in the place of an OP, issue associations, and make choices. The message signatures in this situation are no longer sufficient.

Additionally, if the discovery process is tampered with, the attacker will be able to specify any OP without having to pretend to be someone else.

Furthermore, even MITMAttack is not necessary if the XRDS document is altered in order to compromise the confidentiality of information during the discovery process.

Phishing attacks are a major problem when utilising OpenID. In these types of attacks, the phisher generates a phoney RP that closely resembles the genuine one and sends the user to it. If the user enters the owned OpenID, they will be taken to another fake OpenID provider page that requests their password. If the user enters the necessary password, the phisher will get the password. By doing this, the phisher may utilise any RP service in place of the legitimate OpenID owner by having access to both the OpenID URL and the password.

The goal of the current study is to offer protection against phishing and other types of assaults.

One-time passwords (OTPs). There are several approaches to robust multifactor authentication in general. These techniques rely on biometrics, smart cards, RFID, steganography, watermarking, or hardware or software tokens [8,9-17].

A specialised device, such as the RSA secure ID token, is utilised in hardware token approaches. However, because of the complexity of this device's distribution, management, installation, and configuration, using this method can be challenging. Additionally, this approach might not be very helpful if multiple cloud applications are set up, each using a different service provider. This is due to the fact that these tokens are often set up for a single application.

A password (the OTP) may be sent through SMS text message, Skype, a mobile app, email, instant messaging, and other means in software-token-based techniques. As a result, there is no need for a hardware token since the authentication may be conducted using something the user currently owns. Because there is no distribution or management of hardware tokens, this method has low management overhead. The transaction is safer since the service provider does not retain any seed.Consequently, a flexible software token can easily take the place of a dedicated hardware token.

Biometric authentication techniques rely on the user's bodily attributes, such as voice, iris, palm print, and fingerprints. Limited mobility, rigidity (may be tied to a particular application), and significant overhead due to costly deployment, setup, and maintenance are drawbacks of this approach.

According to these justifications, OTP is a kind of password that may be used for authentication using hardware or software tokens.

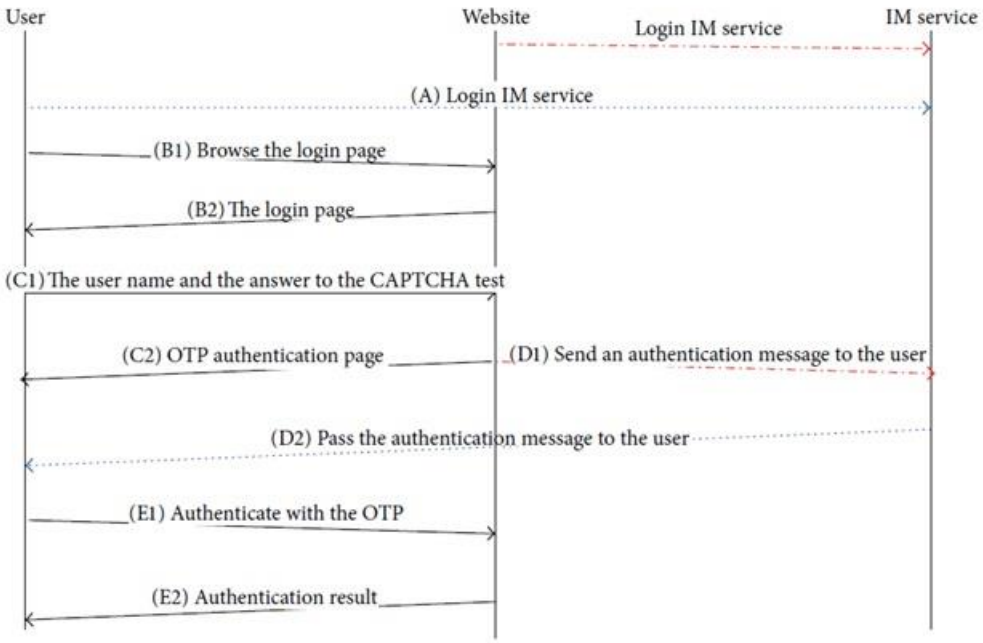

Figure 2: OTP login process [17]

Problem Statement

A method to improve OpenID authentication via the use of an Internet Personal Identification Number (I-PIN) has been put forward by you and Jun in [18].This number, which has been used in Korea, is an exclusive number provided by a user confirmation authority to authenticate users without utilising their personal information.

The method for requesting and receiving an I-PIN is as shown in Figure 2.

(1) The user first asks that an I-PIN be issued with his or her name and residence number from a dependable third party (Principal Confirmation Authority).

(2) The Principal Confirmation Authority verifies the user's identification using their name and resident number, followed by further verification using a certificate of authentication, credit card data, a mobile phone, a face, and other factors (among these).

(3) The Principal Confirmation Authority provides the user's I-PIN when user confirmation is complete.

The generated I-PIN is then used to verify the user's identity throughout the OpenID registration process. Every time this I-PIN is used, the Principal Confirmation Authority is required to confirm user authentication by confirming the accuracy of the user's I-PIN information and transmitting the verification result and principal confirmation information to the identity that requested user authentication.

A TTP is the User Confirmation Authority, a participant in the aforementioned procedure. A TTP is a well-known item used in cryptography that facilitates interactions between parties by examining all of their crucial transactions in light of how easily false digital contents may be created.The participants to this interaction depend on the third party, and since there is already confidence between them in this paradigm, they secure their interactions.

TTPs are typical in all commercial digital transactions in addition to digital cryptographic transactions. The Certificate Authority, which provides Digital Identity Certificates for participants in interactions, is an example of a trusted third party.

A specific kind of third party repository service would be required for transactions that call for third party recordation.The following are some drawbacks of using a TTP.

(i) TTP development, use, and upkeep are exceedingly expensive. For instance, when dealing with digital certificates, a number of factors should be taken into account, including compatibility with the current software platform and tools, tool usability, the strength of the incorporated algorithms (level of security), flexibility, and the development of a platform that accepts all certificates, among others. It is exceedingly expensive to respond to all of these elements and to create and maintain such TTPs.

(ii) For the services they provide, TTPs like Certification Authorities, which issue certificates to different users, charge certain fees. Therefore, subscribing to these services and possessing multiple certificates for various purposes can be very expensive.

(iii) There are restrictions on how an external certification authority may be integrated with an organization's infrastructure.

(iv) There are problems with flexibility when configuring, expanding, and managing the certificates.

Any of the drawbacks of TTPs might be seen as a disadvantage for models that incorporate a TTP as an entity based on the aforementioned justifications.

Additionally, a model that uses OTP to stop phishing is suggested in [19]. The integrity testing of the user's platform has been disregarded in their suggested model since an instant messaging service is utilised (whose application may be installed by any user). While doing so, integrity checking should be taken into account for the security model environment.

Additionally, Madsen et al. [20] discussed federated identity and listed the advantages of federated identity management based on standards in the following manner.

(i) It allows for and makes exchanging user identification characteristics easier for federated organisations' procedures.

(ii) It makes utilising service access requirements for authentication and access authorisation easier.

Additionally, Madsen et al. provided the following examples of current issues and worries in a FIM:

(i) abuse of user identity data via SSO functionality in SPs and IDPs; (ii) identity theft; and (iii) user reliability.

The aforementioned justifications emphasise the current issues with OpenID authentication in [18, 19, 20].The purpose of this work is to develop a secure OpenID paradigm in order to address these issues and limitations.

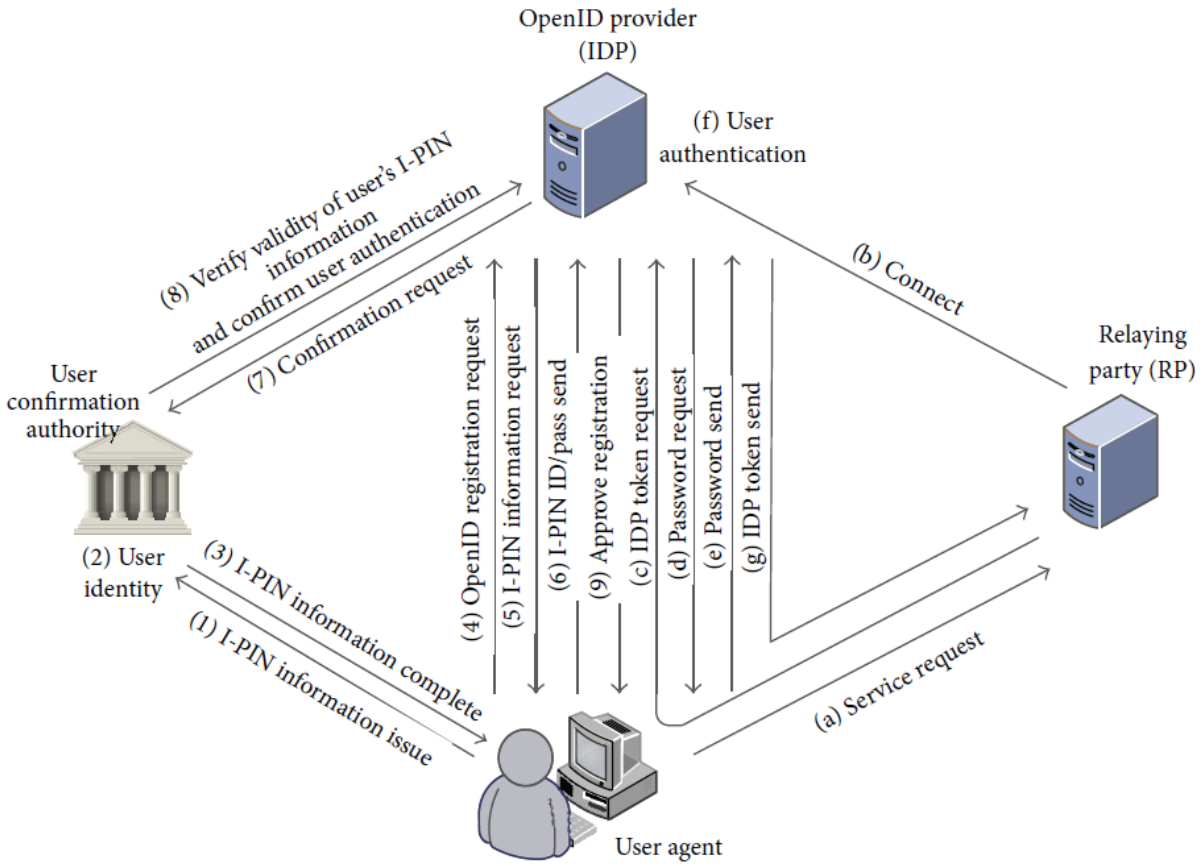

Figure 3: User authentication based on I-PIN [18].

Proposed Model

In our hypothetical situation, we presumptively presume that cloud-based public communications are in use. Users who are registered in the IDP database (at the request of their organisation), depending on their TPM hardware, may execute the OpenID registration and get the OpenID in the event of a migration to private cloud. Additionally, an application that receives the generated OTP is used in the authentication process. This programme is known as the Instant Message Application (IMA). All users who have successfully registered with an OpenID will get the IMA from IDP. Utilising the user's TPM hardware, this application is activated after receiving the OTP that the IDP sends (The recommended method stages explain the whole procedure).

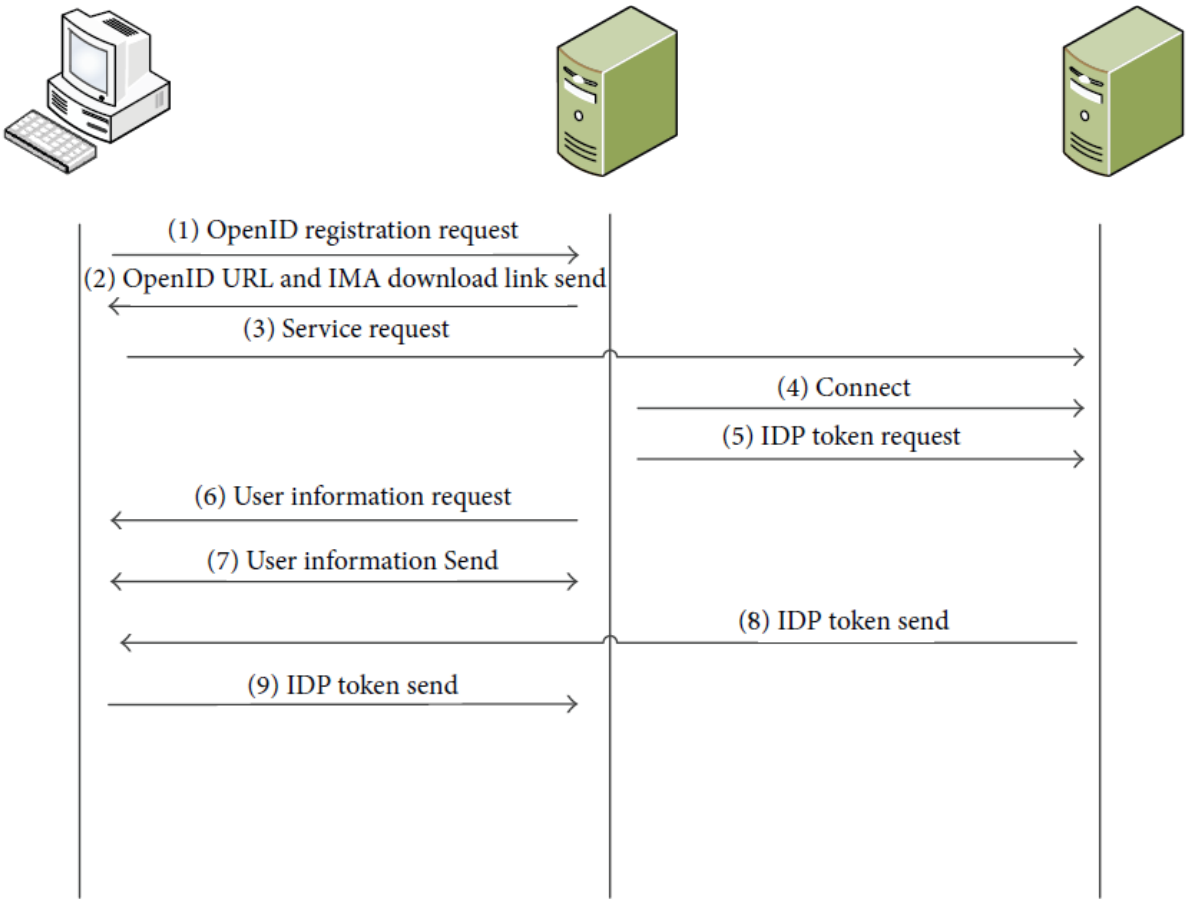

FIGURE 4: Data flow of the proposed model.

a) User Agent asks for an OpenID.The user agent submits a registration request for a PIP account by including a special username, password, TPM key, securitymessage, and personal contact details (address, email address, etc.). As a result, the user will be added to the OpenID user database and given a special OpenID URL.

b) IDP sends the user an IMA download link and an OpenID URL. The user receives an OpenID URL that IDP has created for them. The user is also emailed a link to download IMA. The user gets an OpenID and an IMA download link, after which they download and set up the IMA.

When the user's computer is online and the IMA is opened for the first time, this activation step is carried out automatically. Only when the user requests OpenID from IDP are processes (a) and (b) carried out for the first time. The following are the other stages in the procedure.

1) User Agent asks RP for a service

2) IDP is connected through RP.

3) RP Redirects for User Authentication and Requests Token from IDP via User Agent.

4) IDP Sends SecurityMessage and Requests for OTP and Username.

5) User Agent Sends the Necessary Data.

6) User Information is Requested by IDP for Authentication.

7). Token is sent to RP via User Agent.

Table 1: Feature comparison of existing approaches.

Method	Login on any PC	Security	Price	Trust
My OpenID personal icon	Own	Safe	Inexpensive	Medium
VeriSign and IE	Any	Safe	Expensive	Low
VeriSign and Firefox	Own	Safe	Expensive	Medium
Videoop	Any	Not safe	Inexpensive	Low
Jobber's by SMS	Any	Not safe	Inexpensive	Low
Feng's proposed model	Any	safe	Inexpensive	medium
Secured OpenID model (our proposed model)	Any	Safe	Inexpensive	High

Conclusion

Multitenancy and trusted computing have the ability to address security and trust issues in a federated context. To combat identity theft in the cloud, we have discussed the use of hardware-based activation, OTP, OpenID Web SSO, trustworthy computing, and federated identity management. The study is innovative because it integrates OTP, OpenID, and hardware-based activation, then uses these cutting-edge technologies to suggest a new safe SSO authentication architecture. The study's contribution is seen to be a nexus between trusted computing, cloud computing, and Federated Identity offered by the model, which improves the security and privacy of cloud computing. The deployment of any cloud solution requires trusted and secure identities as well as effective administration of these identities while users' privacy is safeguarded.

To reduce identity theft in the cloud, a suggested architecture for federated identity management based on trusted computing and multitenancy was provided. As a result, we will be able to establish more partnerships based on trust between users, infrastructure components, and suppliers. Additionally, this will make it possible for RPs, IDPs, and individual users to enforce security, trust, and privacy policies. While preventing identity theft is the primary goal, this model can be expanded to address additional federated identity management concerns. Finally, we examined the proposed model's security concerns.

References:

1. Chappell, D.: Introducing windows cardspace. (2006) http://msdn.microsoft.com/en-us/library/aa480189.aspx.
2. Hardt, D.: The OAuth 2.0 authorization framework. (2012) http://tools.ietf.org/html/rfc6749.
3. Recordon, D., Fitzpatrick, B.: OpenID Authentication 2.0 | Final. (2007) http://openid.net/specs/openid-authentication-2_0.html.
4. Sakimura, N., Bradley, J., Jones, M., de Medeiros, B., Chuck, M.: OpenID Connect Core 1.0. (2014) http://openid.net/specs/openid-connect-core-1_0.html.
5. Jones, M., Sakimura, N., Bradley, J.: JSON Web Token (JWT). (2014) http://tools.ietf.org/html/draft-ietf-oauth-json-web-token-21.
6. Google Inc.: Google OAuth 2.0 Client-side. (2015) https://developers.google.com/identity/protocols/OAuth2UserAgent?hl=es.
7. Bray, T.: Verify ID Tokens. (2015) https://www.tbray.org/ongoing/When/201x/2013/04/04/ID-Tokens.

8. M. Alizadeh, W. H. Hassan, M. Zamani, S. Karamizadeh, and E. Ghazizadeh, "Implementation and evaluation of lightweight encryption algorithms suitable for RFID," Journal of Next Generation Information Technology, vol. 4, no. 1, pp. 65–77, 2013.
9. M. Gharooni,M. Zamani,M.Mansourizadeh, and S. Abdullah, "A confidential RFIDmodel to prevent unauthorized access," in Proceedings of the 5th International Conference on Application of Information and Communication Technologies, pp. 1–5,October 2011.
10. H. Taherdoost, M. Zamani, and M. Namayandeh, "Study of smart card technology and probe user awareness about it: a case study of middle eastern students," in Proceedings of the 2nd IEEE International Conference on Computer Science and Information Technology (ICCSIT '09), pp. 334–338, August 2009.
11. A. A. J. Altaay, S. B. Sahib, and M. Zamani, "An introduction to image steganography techniques," in Proceedings of the International Conference on Advanced Computer Science Applications and Technologies (ACSAT '12), pp. 122–126, Kuala Lumpur, Malaysia, November 2012.
12. M. Zamani, A. A.Manaf, and R. Ahmad, "Knots of substitution techniques of audio steganography," in Proceedings of the International Conference on Telecom Technology andApplications, pp. 415–419, 2009.
13. M. Zamani, H. Taherdoost, A. A. Manaf, R. B. Ahmad, and A. M. Zeki, "Robust audio steganography via genetic algorithm," in Proceedings of the International Conference on Information and Communication Technologies (ICICT '09), pp. 149–154, August 2009.
14. J.-H. You and M.-S. Jun, "A mechanism to prevent RP phishing in OpenID system," in *Proceedings of the 9th IEEE/ACIS International Conference on Computer and Information Science (ICIS'10)*, pp. 876–880, August 2010.
15. C.-Y. Huang, S.-P. Ma, and K.-T. Chen, "Using one-time passwords to prevent password phishing attacks," *Journal of 1Network and Computer Applications*, vol. 34, no. 4, pp. 1292–1301, 2011.
16. P. Madsen, Y. Koga, and K. Takahashi, "Federated identity management for protecting users from ID theft," in *Proceedings of the Workshop on Digital Identity Management*, pp. 77–83, 2005.