Indian Journal of Science and Technology, Vol 10(42), DOI: 10.17485/ijst/2017/v10i42/120340, November 2017
ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

The Preliminary Investigation of SSO Protocol for the Suitability of Mission Critical Applications

R. Deeptha[1]* and Rajeswari Mukesh[2]

[1]Department of Information Technology, Hindustan University, Chennai – 603103, Tamil Nadu, India; r.deeptha@gmail.com
[2]Department of the School of Computing Sciences, Hindustan University, Chennai – 603103, Tamil Nadu, India; rajeswarim@hindustanuniv.ac.in

Abstract

Objective: Single Sign-On (SSO) mechanism mitigates the complexity by providing a single set of login credentials for disparate systems. The main objective of SSO in mission critical applications is to provide a most trusted authentication provider while ensuring the secured online service. **Methods:** Exploiting third party identity provider leads Open ID connect to not fit in banking services. Since traditional banking systems refuse to disclose the user-sensitive information to any third party, the conventional models lack in maintaining the consistent revocation model. However, the maintenance of user details at server needs to follow the consistent revocation model at all connected banks. The conventional SSO protocols follow different ways to develop the access control policies to recognize the user identity. However, the banking services need role-based access control policy. **Findings:** In order to provide the secure SSO to banking services, this work presents the extended Open ID connect protocol with additional security features which are more suitable to the online banking systems. The extended Open ID connect provides the role of identity provider to RBI which is a centralized authority to connect all the banks and their user accounts. It exploits the chaotic sequence encryption algorithm to dynamically manage the session, which facilitates the common revocation model. This restricts the impersonation, modification, and eavesdropping attacks while accessing the online banking services with SSO mechanism. The request identification and validation using random user ID in every subsequent page access ensures the security of the online transaction. Moreover, the request identification and validation using random user ID in every subsequent page access ensures the security of the online transaction. **Application/Improvements:** The proposed model tightens the security of Open ID connect with the support of Chaotic sequence encryption, common revocation model, and request identification. It successfully extends the usage of SSO to mission critical applications.

Keywords: Banking Service, Chaotic Sequence Encryption, Open ID connect, Revocation, Identity Provider

1. Introduction

Nowadays, mission-critical applications like online banking applications are widely used among the internet users. The proliferation of mission-critical computing copiously increases the financial transactions by either individuals or the business organizations. Mission-critical applications are highly sensitive in which bit-level interruption also poses the byte-level deleterious impact on the system performance. In addition, due to the numerous and ubiquitous usage of the mission-critical applications, security feature is the pivotal to protect the credentials of each user. The mission-critical application permits the users with the secured login session that needs the user ID and its corresponding password to access their online service. Retaining several credentials in client-side and separately providing the authentication for these credentials in server-side are arduous tasks when the same user has the online account on multiple applications with a different set of login credentials. Single Sign-On (SSO)[1] mechanism

*Author for correspondence

plays a prominent role in resolving the difficulties of handling the credentials when the user relies on the multiple sign-on mechanisms. It enables the users with a single set of user ID and password to access the disparate systems, which facilitates beneficial offers including remembrance of a single login credential and simple system maintenance.

Owing to the use of a centralized set of login credentials for disparate systems, providing the service with secure authentication is a major concern in mission-critical applications. Security Assertion Markup Language (SAML), Central Authentication Service (CAS), Lightweight Directory Access Protocol (LDAP), OAuth2, and Open ID Connect are the existing SSO authentication protocols[2]. These existing SSO protocols are inappropriate for secure banking operations since SSO requires the system to provide the mission-critical data to the third party to maintain the single user ID and password for multiple applications[3]. Moreover, the banking sectors are reluctant to disclose the sensitive information to any third party. Hence, enhancing the SSO authentication protocol without disclosing the confidential information is essential for online banking systems. The proposed system prefers the one authentication enabled Open ID to connect protocol rather than exploiting the computation-intensive SSO protocols. To cope with the secured banking systems, the proposed system enhances the traditional Open ID connect with the extended authentication method.

2. An Overview of Mission Critical Applications

Mission critical application plays a crucial role in the survival of the business activities, including retail or banking systems, airline reservations, border security, and logistics. A mission-critical system widely supports organizations' front office operations while ensuring the secure, scalable, and reliable computing environment. Evolving mission-critical applications require middleware free high-performance security requirements, and well-defined mainframe access. If there is any interruption or failure in the mission-critical system, it significantly creates harmful impact, including the reputations and financial implications on both the customers and organization point of view. For instance, when a customer attempts to buy a robust fund or transfer money, the shutdown state of the server system is not tolerable at that time of such transactions. Moreover, in many IT organizations, process control servers and database systems are mission critical systems. In this case, the potential loss of functions such as faulty hardware or power shortages retreat the business standards as well as considers the system to be mission-critical to the daily operations.

2.1 Risks of Multiple Sign-on in Mission Critical Applications

As an emerging information technology, the mission critical applications such as banking and financial sectors have become incredibly popular in accessing the online services. Accordingly, more than millions of internet users exploit the online banking services to easily perform the commercial as well as business activities in the distributed environment. The different organizations maintain the unique features to create the account registration for each user in a different manner. This uniqueness of multiple organizations leads the following risks in both the user and server side.

- A separate set of user ID and password for sign-on to multiple accounts increase the burden of remembering all the credentials of the user, which is almost infeasible.
- The password fatigue imposes the users to write down their credentials for future remembrance, which reduces the security of maintaining the sensitive information.
- To effortlessly recall their password, the user creates the easily predictable passwords by the third party in which predictable password is regardless of any special characters and complex symbols with the maintenance of separate passwords for each account.
- It is expensive in terms of global IT maintenance cost when the authentication requires verification of every password of the same users' identity in different accounts.
- While maintaining the distinct set of credentials of a single user on different applications, each server needs to separately manipulate the security for the same user, which leads the system to consume more resource in terms of processing and storage cost.

2.2 Advantages of Single Sign-On (SSO) in Mission Critical Applications

In the digital world, users greatly access the multiple systems to perform their day-by-day business activities.

Due to the rapid increase in applications, each user needs to maintain multiple credentials, which in turn has the possibility to forget the credential information. It also creates the inconvenience for the server systems in storing, securing, and processing the multiple credentials for each identity from the perspective of a centralized controller. For instance, Reserve Bank of India (RBI) acts as the centralized controller for all private and public banking sectors. To overcome this inconvenience, the current research focuses on the SSO methods averting the potential confusion in handling the multiple user IDs and passwords for different applications. SSO enables the users to access the multiple services by only exploiting a single set of login credentials. It facilitates both the user and the developer in terms of mitigating the burden of remembering multiple passwords and cost of managing the multiple credentials respectively. SSO method creates a greater impact on the mission-critical applications by managing the multiple sign-on methods. In the case of mission-critical systems, securely authenticating the SSO is paramount rather than merely providing the authentication for many logins. The SSO is the centralized gateway for all the integrated applications rather than providing the separate gateway for each application. If the adversary hacks the SSO credentials, the adversary has the feasibility to access the entire integrated applications in the centralized system. Hence, providing the high secured authentication is essential for SSO in a mission-critical system.

3. Common SSO Security Features to Enterprise and Mission Critical Applications

The security systems seek to standardize the communication of cryptographic secrets between a user and an identity provider in enterprise and mission-critical applications using common security features.

The common security policies are as follows.

Authentication: Authentication allows each user to login into the enterprise applications using secret credentials such as username and password. The traditional encryption algorithms generate the encrypted passwords of users and store them in their database. The attackers can identify the algorithm which is used for encrypting the credentials if they spot any loopholes in the system. Consequently, the hackers get the chances to maintain the lookup table of all possible combinations of the password using the hacked encryption algorithm. Moreover, if the user chooses any dictionary word as a password, the hackers can easily crack it using a lookup table by maintaining an encrypted password for each dictionary word. This vulnerability can be limited by introducing a strong password with numeric values and special characters.

Authorization: Authorization is a process that allows authenticated clients to access a resource by determining whether the client has access rights to the system. The authorization process specifies the access rights to various resources regarding data security. For instance, the security system normally authorizes a legitimate doctor to access patient records. Moreover, the access policy is also formalized as control rules in a system. During authorization, the system approves the access requests from an authenticated user, only when the user ID satisfies access control rules. In banking sectors, the authorization method implements fine-grained access control by providing role-based access. Access manager plays the role of the centralized control center and enforces the access control policies.

Integrity: It ensures that the data does not change during the processes of storage and retrieval. If any change occurs in the data that is sent between the user and relying party, it is likely to be a failure of security. The Transport Layer Security (TLS) protocol provides data integrity between two communicating parties.

3.1 Additional SSO Security Features to Support Mission-critical Applications

Due to the rapid development of mission critical applications that offers the users to access the applications quickly at anywhere and anytime through online, the related financial services especially banking encounter enormous challenges in terms of security. Comparing to other enterprise applications, the banking sectors need enhanced security features due to the involvement of financial aspects which create the direct impact on users. Therefore, the core security features are required in banking services to effectively address the security issues and

offer the defense against hacking activities. Some of the important security features that need to be more focused on mission-critical applications are as follows:

Authentication: Even though, the security vulnerabilities can be limited by introducing a strong password with numeric values and special characters, the hackers are able to create the lookup table regardless. Thus, the traditional authentication methods are not adequate to securely access the banking applications. To secure the user's credentials, most of the banking applications employ salted password hashing method. The salted password hashing method generates a random number for every password and inserts it into the actual password before it is encrypted. Thus, it creates the different encryption values and restricts the possibility of look-up table maintenance.

Request Identification: In addition to the common SSO security features, the financial applications need to execute the authentication for user request also to prevent any session hijack attacks. The request authentication method immediately generates the unique Request ID after validating user credentials and maps it to the customer ID in the persistent storage or server cache. The same value returned to the portal client is appended to the Request ID for the subsequent requests. The request filter receives and validates the Request ID sent by the client against the unique Request ID stored in the server cache. If both of them matches, the request is allowed, then another ID is generated and sent to the client. This process is continuously repeated for every request to enhance the security measures at every step. Due to these subsequent request validations, the session hijacking is not possible in banking applications.

Interoperability: The banking services are highly committed to respond to the credential security of users and to establish Interoperability than other online enterprise applications. Exploiting SSO mechanism in ultra-high security environments such as online banking applications require a highly trusted third party as an Identity Provider (IDP) for maintaining the sensitive information of all banking sectors securely. Even though the Reserve Bank of India (RBI) is considered as an IDP, the separate authentication process between IDP and relying party must be considered to ensure the interoperability. Several issues such as repudiation and revocation affect the secure interoperability in a banking environment.

Repudiation and Revocation: The digital signature generated in the online banking scheme requires authenticity, enforceability, non-repudiation, and integrity to be valid. In banking sectors, the digital signature provides the evidence for non-repudiation capability of beneficial in which the bank identifies whether the beneficiary has the possession of the required asymmetric private key or the transaction amount has been tampered. Moreover, the revocation supports to validate the sender and receiver using several models such as receivers signature validation, credential validation, validation of senders signature on both the transaction and the Certificate Revocation List (CRL) managed by the certificate provider. Every bank has the chance to build own revocation models based on their validation requirement.

4. Review of SSO Protocols

Kerberos[4] is the widely used key distribution protocol in providing the centralized SSO environment, which resolves the security threats such as information leakage and alteration, and hacked passwords. It is likely to lead the adversary to impersonate the data and security tokens via simple network related attacks. To overcome these security breaches in Kerberos, the existing systems[1] employ the Lightweight Directory Access Protocol (LDAP), Security Assertion Markup Language (SAML), and Open ID Connect protocols. The LDAP API allows the users to access the secured directory and SAML provides the standard ensuring the secure authentication and authorization exchange. Open ID Connect[5] is a decentralized authentication protocol for SSO, which is built directly on top of Open Authorization (O Auth) 2.0 protocols. It allows the users to access the service after performing the verification by an OP through authentication. Open ID Connect is the integration of both Open ID and O Auth 2.0 in which Open ID is an earlier version of Open ID Connect[6]. The OAuth2[7] is a web SSO improving the security performance of O Auth 1.0, which only offers the authorization and lacks in providing the authentication. An approach presents the new Open ID Connect protocol and its design features includes the extension

in the discovery and dynamic registration. Even though recent works extend the Open ID Connect protocol with the novel security features, still there is requirement of the intelligent authentication method due to the vulnerability of malicious attacks. Moreover, these extensions are not suitable for mission-critical applications. Hence, the extension requires the authentication method without disclosing the sensitive information to any third party.

SSO protocols ensure in resolving the interoperability issues in the mission-critical systems such as online banking by providing the federated identity credentials regardless of the re-authentication of the disparate systems[8]. The work[9] analyzes the SAML-based security model in terms of security flaws and feasible attacks on the protocols while maintaining the single login credentials for e-banking account. To provide the secure e-banking system, the work in[10] presents a solution with the integration of core banking system with SSO. An extended Central Authentication Service (CAS) SSO solution[11] resolves the security constraints in mounting the existing applications and providing the complex permissions assignment by optimally integrating the enterprise systems. The work[12] exploits the chaotic signal to extract the hidden information based on the time-delay system, which is used to encrypt and decrypt the information in the communication systems.

4.1 Comparison of Important SSO Protocols for Mission Critical Applications

In recent years, research works focus on exploiting the SSO mechanisms to facilitate the users to access all integrated applications in a secured manner. The SSO protocols such as Kerberos, O Auth 2.0 protocol, LDAP, and Open ID are introduced to resolve the security threats. This section describes the most notably preferred existing SSO protocols such as SAML, CAS, and Open ID Connect protocol for the mission-critical applications. The SAML SSO protocol is a framework that enables the exchange of security and identity information among different organizations with different security domains such as IDP and relying party. It has specific rules and syntax for exchanging the information. However, it is not a solution for granting access and enforcing identities in mission-critical applications. The primary aspect is that the SAML depends on assertions about identities. It is assumed that an IDP is creating an accurate assertion and it is responsible for authenticating users, retaining user identities, and determining privileges. The assumption of SAML is that an IDP executes all of the security functions and does not standardize the authentication and authorization in mission-critical applications. Thus, the SAML does not support to verify the quality of IDP and to assure that the IDP is compliant with security features in banking applications. In addition, the Open ID Connect protocol is a new REST full model based on SAML protocol with the desired changes according to REST architecture. The Open ID Connect is designed for the consumer-to-social-network scenario, and it does not fulfill the solid security features enforced by mission critical applications. The extensions in its design features such as discovery and dynamic registration are still vulnerable to attacks. Moreover, the CAS SSO protocol provides a trusted way to authenticate a user through centralized architecture using a single server. In the CAS-based application system integration, the CAS server maintains only one overall user table which manages passwords to authenticate users. Thus the permission assignment task for all systems is complex. Another issue is the inflexibility of CAS when mounting a new application into the system. As the CAS protocol includes CAS Authentication Filter, Validation Filter, and Http Servlet Request Filter, it results in configuration information cumbersome and loss of system generality. Some of the related factors of Open ID Connect, SAML, and CAS are listed in Table 1. From the table, it is observed that the Open ID Connect easily supports to further extension for mission-critical services, comparing SSO protocols.

4.2 Advantages of Open ID Connect in Mission Critical Applications

Nowadays, Open ID Connect has become a standardized SSO protocol supporting identity management service[1]. It is a decentralized authentication protocol, which is more advantageous to alleviate the disclosure of user-sensitive information by identity oriented attacks in mission-critical applications. Open ID Connect plays a major role in rectifying the limitations of existing SSO authentication protocols with the capability of authentication, interoperability, authorization, and secure attribute transmission[13]. The functionality of Open ID Connect in mission-critical applications is discussed as follows using the security features of interoperability, revocation, and request identification.

Table 1. Comparison among SSO Protocols

Comparative Factors	Open ID Connect	SAML	CAS
Purpose	Providing authentication layer on O Auth	Exchanging authentication and authorization data	Centralized authentication
Implemented Between	Relying party and IDP	Relying party/IDP	Relying party and IDP
Technique	URL and HTTP	XML-Enc	Data Encryption Standard
Possible Threats	Phishing, Man-in-middle, and replay attacks	XML signature wrapping attacks	Phishing and steal service ticket
Usability	Need adaptability	Complex	Inflexibility
Token Type	Plain text messages	SAML assertions	Ticket-granting cookie
Support on Security Features	Authentication, authorization, and integrity	Authorization	Authorization and integrity
Support on Banking Services	Interoperability, revocation, and non-repudiation	Revocation and non-repudiation	Revocation and non-repudiation

To ensure the protected service to the end-users, Open ID Connect employs three types of flows such as authorization code flow, implicit flow, and hybrid flow. Authorization code flow exploits two types of tokens such as access tokens and ID tokens. By exploiting these tokens, Open ID provider provides the authenticated service to the end-users by generating the authorization response. In authorization code flow, the Open ID provider generates the access and ID tokens when only recognizing the secret information from relying party. An advantage of authorization code flow is that there is no token available at the user side, which restricts the illegal access of the user agent. Even though, the use of refresh tokens reduces the performance and management overhead; there is a possibility to launch session hijack attack in authorization code flow. The hybrid flow sends the tokens directly to the relying party without using an authorization code. In Hybrid flow, the authorization server and token endpoint generate the authorization token and ID token respectively, but it directly returns the tokens to the end-user. It lacks in achieving better interoperability and revocation. To tighten the digital security from disclosing the credentials, Open ID Connect needs to be enhanced with the features of request identification and interoperability to support mission-critical applications.

5. Technical Challenges in Adopting Open ID Connect for Banking Applications

Multiple sign-on mechanisms burden the users' day-to-day activities in terms of protecting the credentials from the adversary when externally storing the sensitive information and also leveraging the credential management with high cost. Most of the organizations employ the Open ID Connect to provide the secure authentication and authorization service due to its simple identity layer and adoption to the growing community. There are two issues related to the SSO server and operation.

5.1 Challenges Related to the Server

- Traditional banking systems refuse to disclose the user-sensitive information to any third party, hence providing the role of IDP in SSO protocol is an arduous task in banking services[14]. There is only one way to perform the role of IDP i.e. Using RBI. RBI is a centralized authority to connect all the banks and their user accounts. Thus, the communication between bank server and RBI must be more secure, due to the handling of financial transactions.

- Each bank can follow any of the revocation models[15] such as sender validation with or without evidence, receiver validation, and sender validation with cosigning to build its revocation model. However, maintenance of user details at RBI needs to follow the consistent revocation model at all connected banks.
- In addition to the user credential, the bank credentials also must be managed. During user login, bank to RBI communication needs to handle the interoperable credentials[16]. The additional technical challenge is to issue certificates to banks for avoiding liability.

5.2 Challenges Related to the Operation

- There are several operation-related issues such as interoperability. During the communication between relying on party and RBI, a cross-site forgery attack often redirects the users to other sites and accesses their bank account without the intention of the users [17].
- The existing SSO authentication methods for enterprise applications are not always consistent due to the maintenance of unique user ID for each user's identity. Hence, to support online banking services in terms of non-repudiation and revocation, global maintenance of user profile at RBI requires the additional attributes and token update for every session during the user registration.
- The enterprise applications follow different ways to develop the access control policies to recognize the user identity. However, the banking services require role-based access control policy. For instance, the bank manager is only allowed to access the transaction details of bank users, but the user can access the details of balance inquiry, transaction, and statement.

6. Extended Open ID Connect for Mission Critical Applications

The proliferation of disparate systems and the popularity of the mission-critical applications pose the challenges for end-users and the system administrators. The major constraints of multiple sign-on describe that the end-users meet the frequent tracking of multiple electronic identities on different applications and the system administrators need to manage the security for those disparate systems. To resolve these constraints, the existing works exploit the Open ID Connect SSO authentication protocol for mission-critical applications. However, the existing Open ID protocol is not suitable for the banking systems as they are not efficient in handling high-risk financial transactions. Thus, extending the Open ID Connect is crucial to strengthen the security of user-sensitive information in banking systems through SSO authentication method.

The proposed system presents the banking system with secured SSO authentication by dynamically encrypting the user-specific login credentials, creating a common revocation model, and request identification which is shown in Figure 1. It employs the chaotic sequence encryption and decryption mechanisms to securely create the dynamic session IDs for each user while accessing the online banking service. It performs the authentication and authorization by establishing the SSL/TLS communication channel between the end-user and the RBI. The proposed approach incorporates three major phases including Dynamic encryption based robust authentication, common revocation model, and request identification.

Dynamic encryption based robust authentication: The proposed approach focuses on providing the secured service to the end-user. When the end-user enters into the homepage of a bank website, the RBI creates the first level session ID for each user, which mostly depends on the IP address and several fields such as a random number, and session count. To avoid the impersonation attack, the proposed model takes into account the timestamp information to create the dynamic register ID for each online bank user instead of exploiting the IP address, and CAPTCHA, which averts the malicious login of the legitimate user. RBI validates the received information of the user with the credential information stored in the database in which RBI maintains the control of RBI. It redirects the end-user to the corresponding banking site by providing the access tokens and ID tokens.

Common revocation model: After the completion of authentication, the end-user initiates their session to perform the desired operation on the bank site. To avert the modification attack, the proposed model generates the time of website accessed by the user within the session, and the timestamp is encrypted using the chaotic sequence encryption method at RBI. The relying party redirects the request with

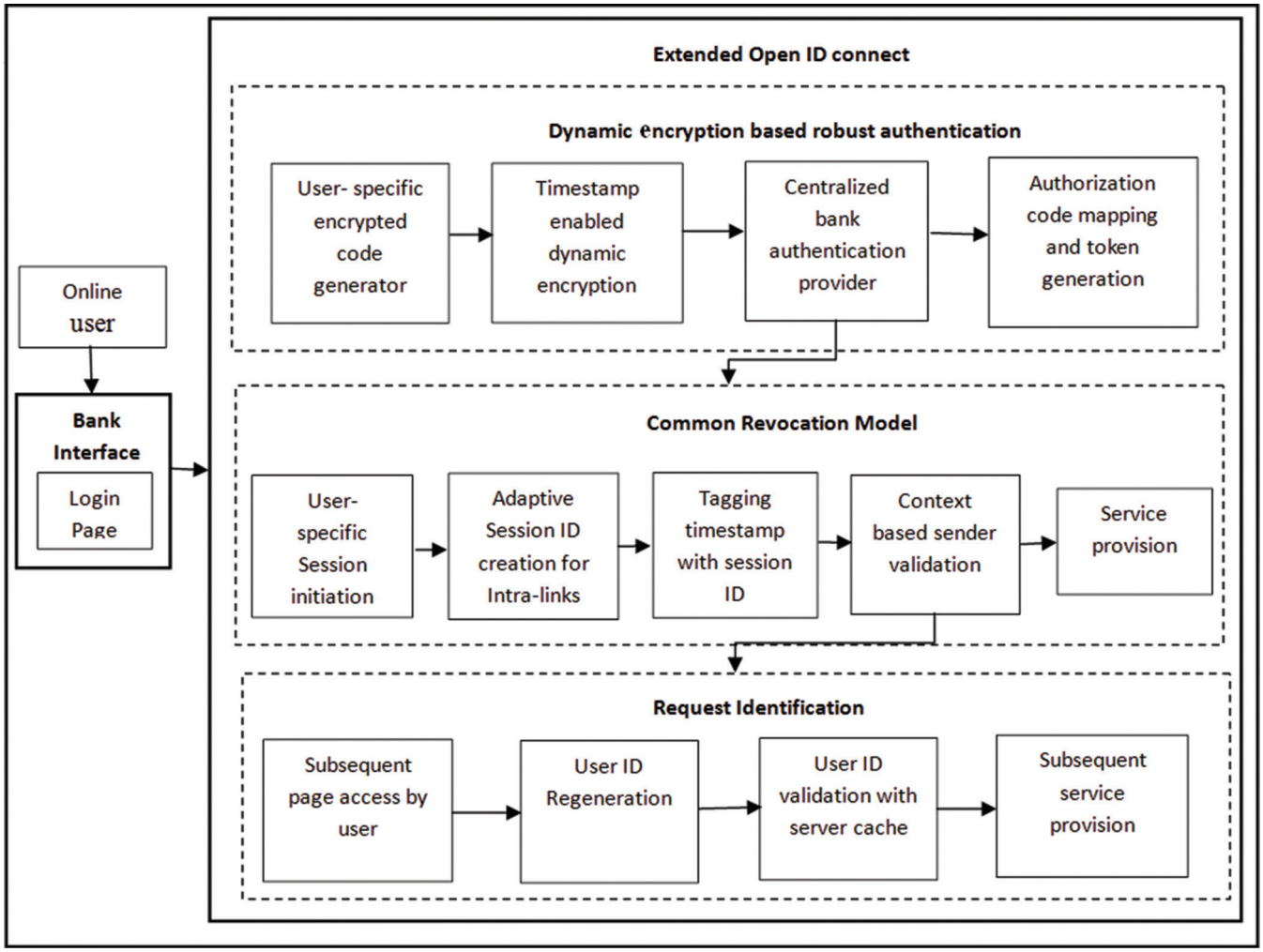

Figure 1. Dynamic SSO authentication method for mission-critical system.

an encrypted timestamp to the user. The user again encrypts the timestamp using the chaotic algorithm. If the adversary modifies the credentials entered by the user, the RBI identifies the attack by decrypting the double encrypted timestamp. In the case of any changes in the timestamp, the RBI informs the user through either e-mail or mobile phone. Finally, the proposed model provides the secured SSO authentication service by the extended Open ID Connect protocol.

Request Identification: In addition to authentication, each request generated by the authenticated user must be validated. This prevents the session hijack even after the secure login. In extended Open ID Connect, the user credentials are stored in a server cache during registration. For every login, the user credentials are authenticated, and the user is allowed to access their pages by generating a request. In the case of accessing the subsequent pages, the unique ID (UID) is generated and appended to the request packet along with the request ID. The same UID is returned to the server cache, which is appended in the subsequent request packets. The request filter validates the UID, when receiving the request packet using server cache. The request is processed when the UID matches, otherwise it is dropped. The enhanced Open ID Connect repeats the same process for every request and secures the high-risk financial transaction.

The flow diagram of authentication, revocation, and request identification process is shown in Figure 2. The enhancement of Open ID Connect with the features of better interoperability and request identification ensures the possibility of using the SSO protocol in secure financial transactions.

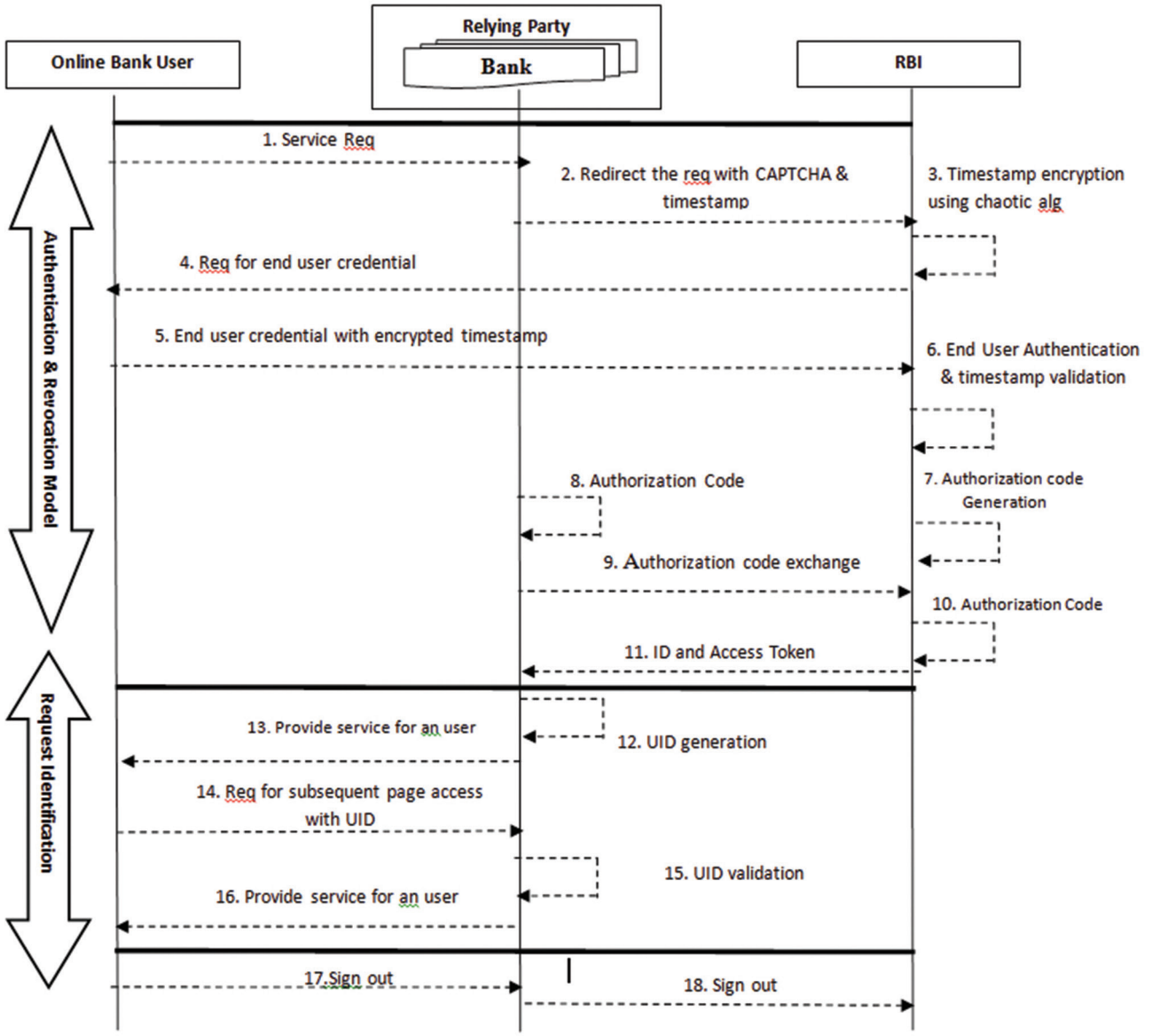

Figure 2. Flow diagram of extended open ID connect in mission-critical applications.

7. Conclusion

This paper presents the extended Open ID Connect protocol for the mission-critical applications. Open ID Connect protocol is the web-based distributed SSO protocol providing both the authentication and authorization services. This paper analyzes the existing SSO authentication protocols, and its challenges while adopting the traditional SSO model to the banking systems. This paper discusses the additional SSO security features needed for banking applications. The proposed model enhances the existing Open ID Connect by tightening the security mechanism with the support of chaotic sequence encryption, common revocation model, and request identification. The proposed model prevents the impersonation, modification, and eavesdropping attacks by misleading the adversary through dynamically encrypting the predictable information such as a user ID and password, and timestamp.

8. References

1. Radha V, Reddy DH. A survey on single sign-on techniques. 2nd International Conference on Computer Communication Control and Information Technology. 2012; 4:134–9. Crossref.
2. Pashalidis A, Mitchell JC. Taxonomy of single sign-on systems. Springer 8th Conference on Information Security and Privacy. 2003; 27(27):249–64. Crossref.
3. Li W, Mitchell CJ. Analyzing the security of Google's implementation of Open ID Connect. Detection of Intrusions and Malware and Vulnerability Assessment. 2016; 9721:357–76.
4. Mukhamedov A. Full agreement in BAN kerberos. Proceedings of IEEE Workshop of the 1st International Conference on Security and Privacy for Emerging Areas in Communication Networks; 2005. p. 218–23. Crossref.
5. Open ID Connect Core 1.0 incorporating errata set 1 [Internet]. [cited 2014 Nov 08]. Available from: https://openid.net/specs/openid-connect-core-1_0.txt.
6. McIntyre JB, Luterroth C, Weber G. Open ID and the Enterprise: a model-based analysis of single sign-on authentication. Proceedings of IEEE Conference on Enterprise Distributed Object Computing; 2011. p. 129–38.
7. The O Auth 2.0 Authorization Framework [Internet]. [cited 2012 Oct]. Available from: https://tools.ietf.org/html/rfc6749.
8. Benson G, Chin SK, Croston S, Jayaraman K, OldSer S. Banking on interoperability: Secure interoperable credential management. Elsevier transaction on Computer Network. 2014; 67:235–51.
9. Grob T. Security analysis of the SAML single sign-on browser/artifact profile. IEEE Proceedings 19th Conference on Computer Security Applications; 2003. p. 298–307.
10. Abdollahi A, Afzali M. Towards securing e-banking by an integrated service model utilizing mobile confirmation. Research Inventy: International Journal of Engineering and Science. 2014; 4(9):26–30.
11. Wang Z, Guo Y, Tang W, Xu Y, Feng B, Hou Q. Research and implementation of single sign-on in enterprise systems application integration. Springer International Conference of Young Computer Scientists Engineers and Educators; 2016. p. 157–70. Crossref.
12. Prokhorov MD, Ponomarenko VI. Encryption and decryption of information in chaotic communication systems governed by delay-differential equations. Elsevier transaction on Chaos Solutions and Fractals. 2008; 35(5):871–7.
13. Mladenov V, Mainka C, Schwenk J. On the security of modern single sign-on protocols: Second-order vulnerabilities in Open ID Connect; 2015. p. 1–15.
14. Ardagna CA, Damiani E, Frati F, Reale S. Adopting open source for mission-critical applications: A case study on single sign-on. Springer IFIP International Conference on Open Source Systems; 2006. p. 209–20. Crossref.
15. Carnahan LJ, Smid ME. Security requirements for cryptographic modules. NIST Publications; 1994 Jan. p. 1–56.
16. Griffin R, Sankuratripat S. Key management interoperability protocol profiles Version 1.0. Technical Report OASIS; 2010.
17. Zeller W, Felten WE. Cross-site request forgeries: Exploitation and prevention. The New York Times; 2008. p. 1–13.