TELKOMNIKA, Vol.16, No.4, August 2018, pp. 1745~1751
ISSN: 1693-6930, accredited First Grade by Kemenristekdikti, Decree No: 21/E/KPT/2018
DOI: 10.12928/TELKOMNIKA.v16i4.8388

Token-based Single Sign-on with JWT as Information System Dashboard for Government

I Putu Arie Pratama*, Linawati, Nyoman Putra Sastra
Master Program of Electrical Engineering, Udayana University, Jl. P.B. Sudirman, Denpasar, Bali 80232, 0361- 239599, Indonesia
*Corresponding author, e-mail: putuariepratama@unud.ac.id

Abstract

Various web-based information systems are developed by Indonesian government to improve quality of services for their society. It encourages users, generally civil servants, to perform different authentications on used information systems and have to remember credentials. Account management of the users poses another challenge for administrators. Single Sign-On (SSO) can be the solution by providing a service of centralized authentication and user account management. This study applies a token-based SSO architecture and uses Json Web Token (JWT) to grant permission authorities, since JWT can provide a claim process between 2 parties. Additionally, the built-in dashboard lists associated information systems to facilitate accessing for the authenticated users. This study will discuss JWT implementation on the dashboard of government information systems that implements SSO, which will generate the permission authorities securely for connected information systems on SSO.

Keywords: *Single sign-on, Json web token, Government information system*

1. Introduction

The development of information communication technology (ICT) encourages the government to implement e-government by exploiting the opportunities provided by ICT in improving the quality of public services to their society [1]. According to the opportunities created by ICT technologies, many developing countries invest in ICT to implement e-government systems [2] as e-government is able to bring efficient and to solve the problems in administrative processes as well as public operations [1],[3],[4]. Therefore, the utilization of information systems is done in various components, so various information systems are also developed [3]. Information systems that base on web technology becomes a preferred solution chosen by government which offers interactive capabilities in providing digital information or services [5]. The developed information systems have an authentication process, so civil servants that generally as users have to pass through different authentication processes, even in the immediate short of time when they access each system without an authority yet.

An authentication process on web-based applications requires a user to have an account, then a registration process is also required. It also occurs on applications that are built in a single organization of government, which users have to complete registration processes on installed systems. Even though users generally use their self-identity on the registration processes [6] such as name, identity number and email. However, they consider to use strong combination of username and password and even differently for each application in reason of security. As most applications have the authentication process, issues arise hence the users are required to memorize inevitably all accounts along with the users have to perform partial authentication processes. Beside that, management of decentralized user accounts becomes another challenge faced by administrators considering growth of user account data [7].

Implementation of single sign-on (SSO) can unify the authentication processes existing on multiple applications to provide centralized authentication and user data management services. All applications can be accessed simply by performing an single authentication process. Then, the time for authentication processes can be optimized and users also do not have to remember many passwords [7], [8]. The implementation of SSO is also support and compatible to electronic public administration concept as SSO has been experimented in e-government interoperability frameworks [9]. Furthermore, SSOs are able to be applied on multi-

Received December 19, 2017; Revised April 1, 2018; Accepted June 11, 2018

domain information systems, which the multi-domain authentication is required along with the comprehensive development of network applications as happened in the government [10].

According to architecture of token-based SSO, a user will be generated a token once the user passes authentication process successfully. However, the token contains only information for an authorization of system access without specifying resource permissions that will be assigned to users [11-13].

Json Web Token (JWT) has advantages as a token authorization of SSO, since JWTs have an ability of determining resource permissions when the users access an information system. By exploiting specialty of JWT that can contain entities, roles of users can be sent simultaneously with the authorization in a single token. Beside that, JWTs can represent a set of claims between 2 parties [14], hence a JWT token has an ability to authenticate the sender of the request on the receiver. That is in line with the architecture of token-based SSO.

We compare performance the proposed SSO to a non-JWT based SSO that usually uses an encryption token. We also compare a token of JWT containing roles to a token of JWT without roles and a token of non-JWT. Therefore, we show performance of the tokens and observe effect of inserting roles of users in a token of JWT. Additionally, the proposed SSO has been implemented with another speciality, we build a menu dashboard that provides a list of the information systems to facilitate users access. This paper is outlined as follows; in Section 2, this paper discusses about the design of proposed SSO. In Section 3, discussion about the implementation is done. Section 4 discusses the system testing and result. Lastly, Section 5 concludes the paper.

2. System Design

There 3 phases that an SSO protocol need to be owned to provide a service of centralized authentication. The 3 phases generally operate on 3 roles that are end-user, Service Provider (SP) the so-called Information System, and Identity Provider (Idp) the so-called SSO [11], [15-17]. The first phase is to register and create trust establishment between SSO and the information system, phase 2 is a registration process users on the SSO, and the final phase is the users authenticating on the information via a token created by the SSO [11].

The dashboard of information system for government is built using a token-based SSO architecture. Once a user is authenticated, the user will be generated tokens to access the resources or services on information systems without further authentication. User-accessible information systems have been pre-configured to connect with the SSO in order to use the centralized authentication service [12]. In this study, the generated token will be based on JWT considering capability of JWT that can apply a token claim process between 2 parties. In addition, there is a signature verification process using a secret key in JWT. The verification process will be used to validate the authenticity of tokens sent from a trusted source originally, in this case the SSO is the source in question. Secret keys of registered information systems are created when configuring a bond between the SSO and the information systems via an SSO client module that is written for web-based applications. The secret keys apply a symmetric key concept hence the SSO and an information system will use a same key in generating and verifying the tokens [18]. Beside that, the secret keys are designed unique for every information system as shown in Figure 1.

A token is the critical point of security on the architecture [11], since the token is used to authenticate a user on a information system. As discussed on [19], the proposed SSO architecture implements a dynamic token instead of a static token. According to the concept, this study uses tokens combining dynamic content by including date and time in secret key of JWT signature when the generating process. The JWT also support expiration that will describe in detail in Section III. Therefore, a token is only valid at the specific periodic time, called time-limit token further, and able to prevent a replay attack as discussed on [20] that proposed one-time password as the solution.

A login page, provided by the SSO, will appear if a user has not been authenticated. The SSO will direct the user to an information system immediately after the user successfully passes an authentication on the login page. However, on the dashboard of government information systems, the process will occur if the user accesses the information system directly and the user has been authenticated by passing the main authentication on the SSO as shown in Figure 2.

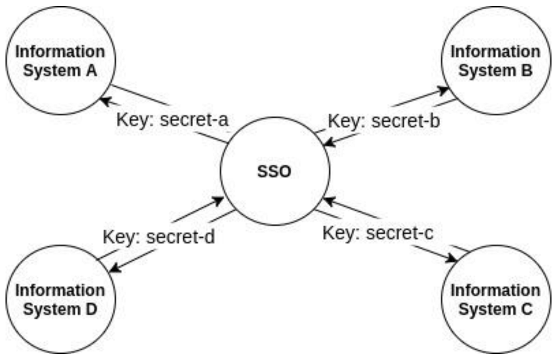

Figure 1. The bond between an SSO and information systems with symmetric keys

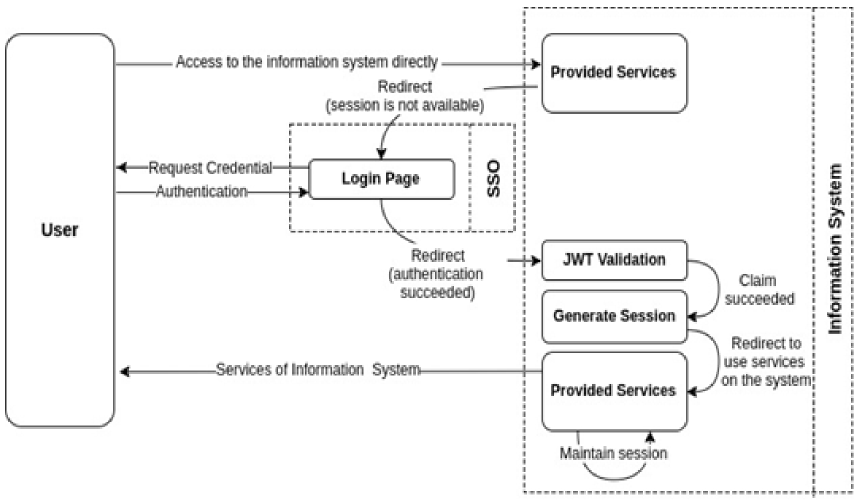
User
Access to the information system directly
Provided Services
Redirect (session is not available)
Request Credential
Authentication
Login Page
SSO
Redirect (authentication succeeded)
JWT Validation
Claim succeeded
Generate Session
Redirect to use services on the system
Services of Information System
Provided Services
Maintain session
Information System

Figure 2. Authentication process when accessing directly to an information system

The dashboard provides a list of information system menus which the permissions are owned by the authenticated users. Therefore, the users do not need to remember the domain of information systems, they simply access the dashboard and pass authentication process to proof permissions of accessing services on each shown information system on the dashboard as shown in Figure 3.

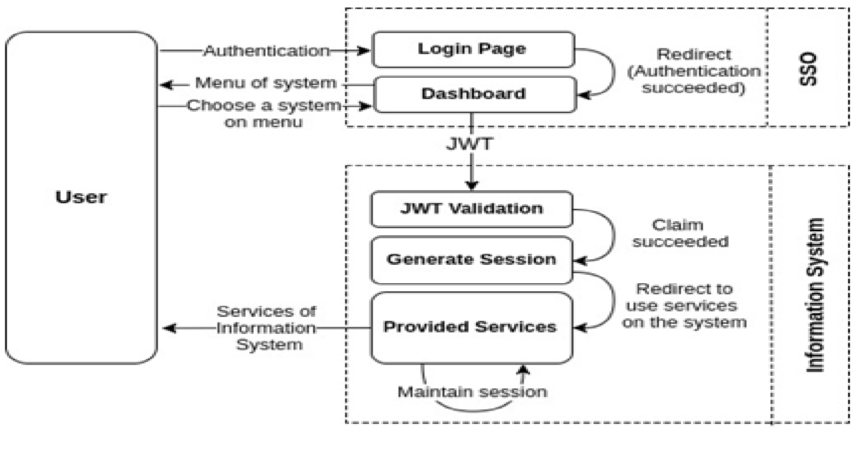

Figure 3. Access an information system via dashboard provided by the SSO

3. Implementation

A JWT consists of 3 json object structures that are encoded and separated by dots (.), ie:

a. Header, there are 2 attributes on the header, one of them is ”type” which in this study will be filled with JWT. Another attribute, ”alg”, is a hashing algorithm used such as HS256 or RSA.
b. Payload, contains an object of claims. The claim is a statement or message about an entity that commonly is a user entity.
c. Signature, used to verify the sender of JWT comes from the correct source and to ensure that the message is not changed [21].

Once a user is successfully authenticated, the dashboard as the SSO will have user information to create a permission authority for each information system differently using token as credential. In the authority process, the SSO redirects and sends the token based on JWT to an information system that is accessed. The JWT which as the user credentials will certainly contain the user entity, the example of JWT structure that is created SSO as shown.

a. Header, the used algorithm is HS256 (HMAC SHA256) and the type is JWT.
```
{
        "typ": "JWT",
        "alg": "HS256"
}
```
b. Payload, contains user information which the entities are such as user ID, identity number (civil servant identifier) and organization ID, roles of user as well as recommended claims listed on *[*14*]* such as iss (issuer), exp (expiration time), nbf (not before), iat (issued at) , jti (JWT ID).
```
{
        "user": {
                "id_user": 1,
                "id_unit": 1,
                "identity": "199305012017011001",
                        "name": “Employee”
        },
        "roles": {
        "1": "Administrator"
        },
        "sub": 1,
        "iss": "http://sso-server.com",
        "iat": 1508505759,
        "exp": 1508509359,
        "nbf": 1508505759,
        "jti": "abcdefgh12345678"
}
```
c. Signature, created by combining header and payload and encoded using HMAC SHA256 with secret key which is configured on the SSO and the client module.
```
signature=HMACSHA256(base64UrlEncode(header)+”.”+base64UrlEncode(payload),
secret)
```

Afterwards, the 3 structures are processed using the standard JWT algorithm and according the JWT creation stages to produce a token that are well-suit and easily passed in HTML and HTTP environment. Below is an example of a JWT that has encoded header and payload as well as been signed with a secret key.

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxIiwibmFtZSI6IiJ9.YPfNDvrY1opL2zEe5K2IGW8c_8OnV6Vpz2EX1A-ebXg

The token as example above is sent by SSO to the accessed information system, the token will be validated in accordance to the standard JWT validation algorithm once the information system receives the token. Beside validating that the token is a valid JWT, validation is also performed based on checking 3 entities as the basis for granting the permissions authority, namely:

a. Based on the nbf (not before) claim, to ensure that the token is claimed after or equal to the date/time listed on the nbf claim which generally contains the date / time the token is created, in this case the token is created by the SSO.
b. Based on the exp (expiration time) claim, to ensure that the token is claimed prior to the date/time listed on the exp claim, the date / time of the exp claim is determined from the addition of the token time to take effect on the date / time of the token. In this study, the token lifetime is set to be valid for an hour (60 minutes) on the SSO.
c. Based on the signature, to ensure that the token is correctly derived from the SSO. In this study, the key for the signature are generated by combining the creation date, the IP address of the information system and the secret key of the SSO bond with the information system (e.g: secret) that has been configured.

secret signature=md5(date + secret + IP address)

JWT validation stages for permissions authorization can be described with a flowchart as shown on Figure 4. After the token is validated, the information system generates a session according to a user's credential contained on the token. The session will then be managed by the information system as long as the user uses the provided services. Thus, the JWT validation process will only be performed when the session of the user is not available yet.

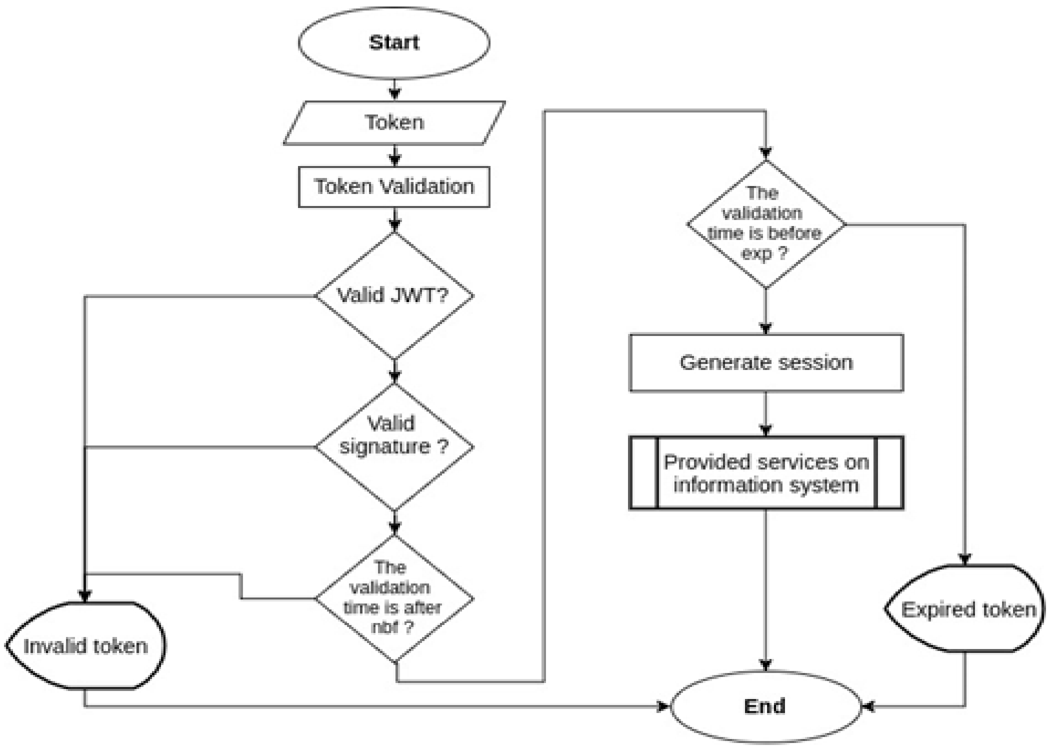
Start
Token
Token Validation
Valid JWT?
Valid signature ?
The validation time is after nbf ?
Invalid token
The validation time is before exp ?
Generate session
Provided services on information system
Expired token
End

Figure 4. Flowchart of JWT validation

4. System Testing and Result

Our SSO has been tested as follow: firstly, we try to insulate a token that is not actually a JWT to ensure the SSO being able to refuse an authorization request with any unknown tokens. Secondly, we try to use different secret key and date contained in secret signature as described in Section III. This test is to ensure that the SSO client is able to handle a token that comes from unknown sender. Thirdly, we try to insulate an out of date token, either after or before the range time allowed. This third test is to ensure that a token is not able to be reused by an unauthorized user in further time by copying a token generated for a registered user. Therefore, the result shows that our SSO can pass all of the tests and handle tested cases by showing informative pages instead of error.

Our SSO also shows better performance than SSO with conventional token as the JWT based SSO can do authorization and send information user as well as roles of the user with only a single token and a single request. In SSO with conventional token, there are at least 3 connections in an authorization process, because the SSO usually uses the encrypted content to be a token as a ticket of authorization will send the ticket (1st connection) to an information system. The ticket will be validated as soon as it receives. If the ticket is valid, the information system will immediately send a request (2nd connection) using web services or API to SSO in order to get detail information and roles of user then SSO will send a response (3rd connection) as the reply. However, our SSO is able to reduce the number of connections by only sending a JWT based token, which the token is used to authorize a user and get the detail information as well as roles of the user without any further connections.

The effect of inserting roles to a token of JWT in our SSO shows the roles affecting to the token length significantly. Considering each role has different permissions, a user may have more than a single role. We assume 4 roles to compare i.e: verificator (verify requirement of applications), scheduler (manage schedules of surveys), surveyor (notify survey result), and survey verificator (verify survey result). Table 1 shows length of tokens generated in our comparison. By the length, the generated tokens of JWT remain longer than non-JWT significantly, even without a role. Therefore, our SSO consumes larger data size in connections considering amount of data size needed by the JWTs is bigger than non-JWTs.

Table 1. Comparisan of Tokens

Token	Contained Information	Length	Bit (UTF-8)	Byte (UTF-8)
JWT	Detail information of user without a role	356	2848	356
JWT	Detail information of user, a role of surveyor	389	3112	389
JWT	Detail information of user, 2 roles of verificator and surveyor	413	3304	413
JWT	Detail information of user, 3 roles of verificator, scheduler, and survey verificator	448	3584	448
Non-JWT (Hash)	Encrypted secret key	32	256	32
Non-JWT (AES 128-bit)	Contain date, ip address and ID user, then encrypted using secret key	64	512	64

Despite the large data size, The token of JWT has no hassles to send over the HTTP environment. We have tried to pass the token either in GET request and POST request, then it passed successfully without any further issues.

5. Conclusion

It can be concluded that Single Sign-On (SSO) using Json Web Token (JWT) in token-based SSO architecture can be applied to a dashboard of government information systems providing centralized authentication service and list of authorized information systems to users. Based on JWT, roles of the users can also be controlled on an SSO, this is generally required by web-based information systems to manage page permissions or features provided to the users.

The signature verification process on JWT can be used to ensure that the token sender and receiver have a trusted bond while maintaining the confidentiality of a secret key on both parties. a JWT-based token can not be changed during the submission process, if the information on the token changes, the token will be declared invalid in the recipient. Additionally, tokens will only be declared valid if the token is claimed in the date / time range contained in the token. Thus, the use of JWT in a token-based SSO architecture is considered safe.

The limitation of this study is that there is no discussion about how to manage the session of each accessed information system centrally. An example is when a user session is deleted in one of the information systems either due to deliberate or limited session time, it will delete other information system sessions immediately. Thus, there needs to be further study focused on managing the session on the dashboard of government information systems.

References

[1] Tino Schuppan. E-Government in developing countries: Experiences from sub-Saharan Africa. *Government Information Quarterly*. 2009; *26*(1): 118–127.
[2] F Mohammed, AI Alzahrani, O Alfarraj, O Ibrahim. Cloud Computing Fitness for E-Government Implementation: Importance-Performance Analysis. *IEEE Access. 2017;* (99): 1–1.
[3] Delfina S and Luis A. Information Systems Interoperability in Public Administration: Identifying the Major Acting Forces through a Delphi Study. *Journal of Theoretical and Applied Electronic Commerce Research*. 2011; *6*(1): 61–94.
[4] M Maureen Brown. Understanding e-government benefits: an examination of leading-edge local governments. *The American Review of Public Administration.* 2007; 37(2): 178–197.
[5] U Sivarajah. Evaluating the use and impact of Web 2.0 technologies in local government. *Government Information Quarterly*. 2015; 32(4): 473–487.
[6] V Beltran. Characterization of Web Single Sign-On Protocols. *IEEE Communications Magazine-Communications Standards Supplement. 2016*; 54(7): 24–30.
[7] V Radha. A Survey on Single Sign-On Techniques*. Procedia Technology.* 2012*;* 4: 134–139.
[8] Y Tie Jun. *Method of Single Sign-on for Independent Web Systems Based on AJAX.* Proceedings of 2013 3rd International Conference on Computer Science and Network Technology. 2013*:* 310–314.
[9] G Mecca, M Santomauro, D Santoro, E Veltri. On federated single sign-on in e-government interoperability frameworks. *International Journal of Electronic Governance.* 2016*;* 8(1): 6–21.
[10] Zengyu C, Qikun Z, Ming L, Yong G, Junsong Z. Multi-Domain Authentication Protocol Based on Dual-Signature. *TELKOMNIKA (Telecommunication Computing Electronics and Control)*. 2015; 13(1): 290–298.
[11] C Mainka, V Mladenov, J Schwenk, T Wich. SoK*: Single Sign-On Security - An Evaluation of OpenID Connect.* 2017 IEEE European Symposium on Security and Privacy (EuroS P). 2017: 251–266.
[12] Tayibia Bazaz. A Review on Single Sign on Enabling Technologies and Protocols. *International Journal of Computer Applications*. 2016; 151(11): 18–25.
[13] X. e. You, Y. Zhu. *Research and Design of Web Single Sign-On Scheme*. 2012 IEEE Symposium on Robotics and Applications (ISRA). 2012: 383–386.
[14] M Jones, J Bradley, N Sakimura. JSON Web Token (JWT). Internet Engineering Task Force (IETF). 2015.
[15] N Naik, P Jenkins. *Securing digital identities in the cloud by selecting an apposite Federated Identity Management from SAML, OAuth and OpenID Connect*. 2017 11th International Conference on Research Challenges in Information Science (RCIS). 2017: 163–174.
[16] F Nie, F Xu, R Qi. *SAML-based single sign-on for legacy system*. 2012 IEEE International Conference on Automation and Logistics. 2012: 470–473.
[17] P Harding, L Johansson, N Klingenstein. Dynamic Security Assertion Markup Language: Simplifying Single Sign-On. *IEEE Security Privacy.* 2008; 6(2): 83–85.
[18] A Ben Charke, M Chabi, M Fakir. Contribution to the Security of the Information System. *TELKOMNIKA (Telecommunication Computing Electronics and Control)*. 2015: 16(1): 154–166.
[19] F Satoh, T Itoh. *Single Sign on Architecture with Dynamic Tokens.* 2004 International Symposium on Applications and the Internet - Proceedings. 2004: 197–200.
[20] Barlian HP, Heru N, Wijaya K. One-Time Password Implementation on Lego Mindstorms NXT. *TELKOMNIKA (Telecommunication Computing Electronics and Control).* 2014; 12(3): 689–694.
[21] M Jones, J Bradley, N Sakimura. JSON Web Signature (JWS). Internet Engineering Task Force (IETF). 2015.