JURNAL INFOTEL
Informatics - Telecommunication - Electronics
Website Jurnal : http://ejournal.st3telkom.ac.id/index.php/infotel
ISSN : 2085-3688; e-ISSN : 2460-0997

Stateless Authentication with JSON Web Tokens using RSA-512 Algorithm

Aldy Putra Aldya[1], Alam Rahmatulloh[2*], Muhammad Nur Arifin[3]
[1,2,3]Department of Informatics, Siliwangi University, Tasikmalaya, West Java, Indonesia
*Corresponding email: alam@unsil.ac.id

Received 09 April 2019, Revised 08 May 2019, Accepted 22 May 2019

Abstract — Today's technology needs are getting higher, one of the technologies that continues to grow now is Web Service (WS). WS can increase service flexibility on a system. However, security at WS is one of the things that needs attention. One effort to overcome this problem is JWT (JSON Web Token). JWT is one of the authentication mechanisms in WS, with a standard signature algorithm, HMAC SHA256, RSA-256 or ECDSA. In this research we will discuss the performance of JWT RSA-512 which is implemented on SOAP and RESTful. Because based on previous research the speed performance of the 512-bit algorithm is better, but it is not yet known if applied to JWT. The test results show that the speed of the JWT RSA-512 token on the RESTful process is superior to 24.69% compared to SOAP. Then the speed of the authentication of JWT RSA-512 tokens, RESTful is superior to 11.64% compared to SOAP. Whereas in testing the size of JWT RSA-512 generated tokens, RESTful is only 1.25% superior to SOAP.

Keywords – JSON Web Token, RSA-512, Stateless, Web Service

I. INTRODUCTION

Technological developments have a major influence on organizations and individuals. One such technology is Web Service (WS). WS is able to overcome the problem of interoperability because it is stateless and works regardless of the platform used by different sources [1].

The benefit of building a WS for business needs is to improve integration and flexibility, in the interest of integrating services, WS works using internet communication with the HTTP protocol so that it supports any application [2] [3]. However, WS's security problems fall into the top 10 vulnerabilities in the security of the Web Service Application Programming Interface (API) which is less protected according to The Open Web Application Security Project (OWASP) [4].

Various ways to reduce threats to security on web services have been carried out in previous studies, including the Claim based Authentication approach, Token based Authentication, Secure with SSL using ASP.NET MVC web API [5]. The use of token-based authentication has also been done in research [6], [7], [8]. Other studies, claim-based authentication ID (identity) [9], SAML Technology [10], and research conducted [11] show that the use of JWT is safer. One step that can overcome this problem is using JSON Web Token (JWT). The JWT defines a simple and independent method using the data format of objects that are transmitted safely and can be verified because it uses a digital signature.

The current RESTful WS authentication mechanism, JWT SHA-256 is still commonly used, so it can be a threat to RESTful WS [12]. Tests have been carried out regarding the performance between JWT SHA-256 and SHA-512. The test resulted that JWT SHA-512 had better performance than SHA-256 [13]. Attacks on this mechanism have begun to develop, such as Scan-based Side-channel [14]. The attack is certainly a separate threat to the use of authentication mechanisms using JSON Web Tokens with symmetric SHA-256 and SHA-512 algorithms. This research will discuss the implementation of the RESTful WS authentication mechanism using the JSON Web Token with the RSA-512 algorithm that uses an assymetric key. Factor implementation of JWT RSA-512 is an alternative choice of algorithm in the implementation of authentication mechanisms using JSON Web Tokens.

II. RESEARCH METHOD

A. *Web Service*

Web service is a system designed to support interactions between systems on a network. Web services to provide services in the form of information to other systems so that other systems can interact with the system through the services provided by the web service. Web services are also interpreted as an interface that describes several operations that can be accessed through a network [13]. Examples of web service architecture include Simple Object Access Protocol (SOAP) and Representational State Transfer (REST). The web service architecture can be seen in Fig.1.

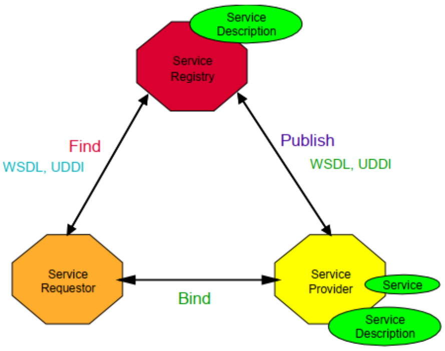


Fig.1. Web Service Architecture

B. *Representational State Transfer (REST)*

The concept of REST is to use resources as components of applications that need to be used or addressed. REST can be explained in five restrictions [9], namely:

a) Resource Identification, meaning the web relies on a Uniform Resource Identifier (URI) to identify resources.

b) Connectedness, meaning the client from RESTFul Service must know the link to find resources in order to interact with the service.

c) Uniform Interface, meaning that resources must be available through a uniform interface with the semantics that defines the interaction.

d) Self-Describing Messages, meaning WS exposes existing resources, RESTFul uses more than one data format (XML, JSON, RDF, etc.) compared to SOAP (XML).

e) Stateless Interactions, meaning that every request from the client is complete and fulfills the need for a request.

C. *JSON Web Token (JWT)*

JWT (JSON Web Token) is a token in the form of a JSON string that can be used to perform authentication and information exchange systems. The small form makes JWT transmitted faster. JWT can be verified and trusted because it has been given a digital signature. The signature used can be by using a secret key (HMAC algorithm) or a public key and private key pair (RSA algorithm) [10].

JWT structure consists of 3 parts separated by dots ("."), Namely headers, payloads, and signatures. The header usually consists of 2 parts, namely the type of token and the hashing algorithm that will be used. The second payload contains a claim. Claim contains data that you want to secure. Signature is formed using headers and payload.

D. *Rivest–Shamir–Adleman (RSA) Algorithm*

The RSA algorithm is an algorithm based on asymmetric key cryptographic scheme. Asymmetric key scheme is a cryptographic scheme using two keys, namely the public key and private key [11].

The stages of the RSA algorithm are as follows:

a) Key generation on the RSA algorithm. The process is as follows:
    1. Select 2 large prime numbers like p, q where p is not equal to q.
    2. Calculate m = p * q
    3. Calculate n = (p-1) * (q-1)
    4. Select e which is relatively prime to n.
    5. Calculate the value of d so that it satisfies (e * d) mod n = 1
    6. (e, m) is a public key for encryption purposes.
    7. (d, m) is a private key for decryption purposes.

b) Encryption. The encryption function is shown in (1),

$$C = X^\wedge{}^{e}\,(\text{mod } m) \tag{1}$$

C is ciphertext from plaintext X encryption.

c) Decryption. The decryption function is shown in (2),

$$N = C^\wedge{}^{d}\,(\text{mod } m) \tag{2}$$

N is the plaintext of ciphertext C decryption.

E. *Related Works*

Vibha Kumari in a journal published in 2015 regarding Web Service Protocol: SOAP vs. REST. It was found that REST is the most widely used WS, but security in the WS is a matter that must be considered [15]. This research will be researched the implementation of JWT RSA-512 on SOAP and RESTful.

Mukhammad Agus Arianto, Sirojul Munir and Khusnul Khotimah from STT Terpadu Nurul Fikri in a journal published in 2016 concerning Analysis and Design of Representative State Transfer (REST) Web Service Academic Information System STT Terpadu Nurul Fikri Using Yii Framework. Information systems are generated by using RESTful WS but have not implemented a security system, so that it makes its weaknesses [16]. In this research, JWT RSA-512 performance testing will be carried out on RESTful WS.

Penidas Fiodinggo Tanaem, Danny Manongga, and Ade Iriani from Satya Wacana Christian University Salatiga in a journal published in 2016 concerning RESTFul Web Service for Transaction Recording Systems Case Study of PT. XYZ. Generated REST that has implemented JWT SHA-256 [12]. However, in this study, the implemented algorithm is SHA-256 which is still very commonly used.

Alam Rahmatulloh, Heni Sulastri and Rizal Nugroho from Siliwangi University in their journals in 2018 regarding Keamanan RESTful Web Service menggunakan JSON Web Token (JWT) HMAC SHA-512. It was found that JWT SHA-512 is better than SHA-256 [13]. In this study, the algorithm used is a symmetric key algorithm. In this research, JWT RSA-512 testing is an asymmetric key algorithm.

Daisuke Oku, Masao Yanagisawa, and Nozomu Togawa in his journal in 2018 regarding Scan-based Side-channel Attacks against HMAC-SHA-256 Circuit Based on Isolating Bit-Transitions Group Using Scan Signatures. This study [17] produces a Side-channel Scan-based attack that has successfully taken the secret key on HMAC-SHA-256 in a relatively short time. This attack is indeed a separate threat to HMAC-SHA-512 which uses a symmetric key (secret key). This research will implement an authentication mechanism using JWT with the asymmetric RSA-512 algorithm

F. *Method*

Stages of research methods that are carried out are literature study, prototype design, trial, and comparison, the analysis and conclusions. The stages of the research method can be seen in Fig.2.

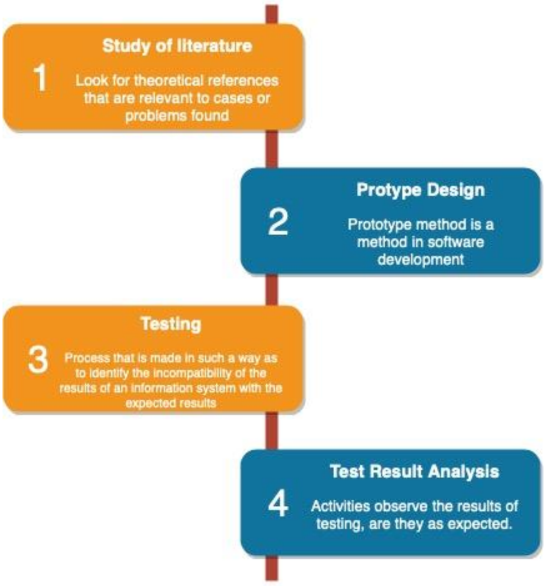

Study of literature
1
Look for theoretical references that are relevant to cases or problems found
Protype Design
2
Prototype method is a method in software development
Testing
3
Process that is made in such a way as to identify the incompatibility of the results of an information system with the expected results
Test Result Analysis
4
Activities observe the results of testing, are they as expected.

Fig.2. Research Method

III. RESULT

A. *Literature Review*

Literature studies have been carried out in previous studies, and it was found that Web Service security is included in the top 10 vulnerabilities [3]. The results of other studies say that REST WS is in the top position in the protocol that is often used compared to other protocols [4]. However, WS security is one of the critical points that must be considered [2]. Therefore JWT is used to cover up deficiencies in terms of security. In previous studies, JWT SHA testing has been carried out using symmetric key [6]. Apart from algorithms with symmetric keys, other algorithms use the asymmetric key. This research carried out the implementation of JWT RSA-512 using the asymmetric key on RESTful WS.

B. *Prototype Design*

d) Analysis of System Requirements

Based on the results of previous studies regarding the implementation of JWT, in this study, the system specifications used were Intel 1.60GHz ~ 2.30GHz Processor, 64bit Operating System, and Postman Tools.

e) Implementation of RSA-512 on JWT

The JWT works like a password, so when a user successfully logs in, the server will give a token. The token is used to make further requests to the server. The primary purpose of JWT is to secure data between systems that will exchange data by making requests by clients that must be included with tokens. In general, the way JWT works can be seen in Fig.3.

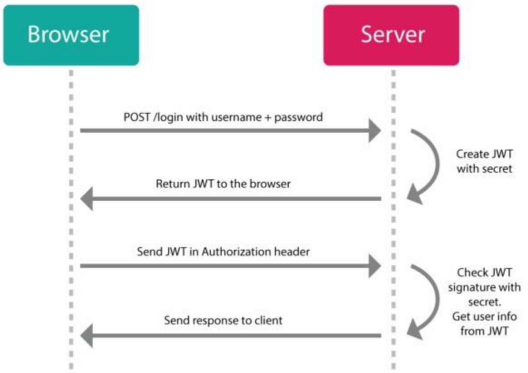

Browser
Server
POST /login with username + password
Create JWT with secret
Return JWT to the browser
Send JWT in Authorization header
Check JWT signature with secret. Get user info from JWT
Send response to client

Fig.3. How JWT Works

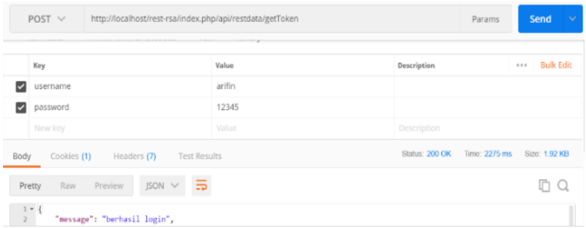
f) Implementation of JWT on RESTful

As in the implementation of JWT on SOAP WS, JWT was tested using the RSA-512 algorithm on RESTful WS. The RESTful WS created has two main functions, namely getToken to request tokens from the server, then the take_barang function that will verify token before the server gives a response.

IV. DISCUSSION

A. *Testing*

This test is done by postman who has a function as the REST Client used to test the REST API that has been created. The parameters to be tested in this test are the speed and size of the token in the generate token process and the speed at the token authentication process. This test is carried out with two processes, namely POST and GET. POST on API / restdata / getToken to send parameters in the form of a valid username and password to generate tokens. Then the GET process is to enter the token and public key that has been obtained in the POST process to obtain data by requesting to / API / goods / take_barang. When making a request, the token is generally sent to the HTTP header. For POST results on API / restdata / getToken can be seen in Fig.4.



http://localhost/rest-rsa/index.php/api/restdata/getToken

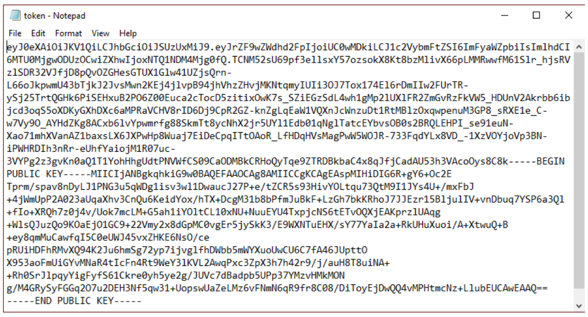
	Key	Value	Description
☑	username	arifin	
☑	password	12345	
	New key	Value	Description

Status: 200 OK Time: 2275 ms Size: 1.92 KB

```
{
    "message": "berhasil login",
```

Fig.4. POST Result

POST on API / restdata / getToken with the username arifin and password 12345, the resulting response is a token and public key. The process of parsing data in the POST process to API / restdata / getToken takes 2275 ms. The size of the token generated can be seen in the token.txt file, which is a file created based on tokens and public keys generated in the JWT generate process. The tokens and the public key generated can be seen in Fig.5.

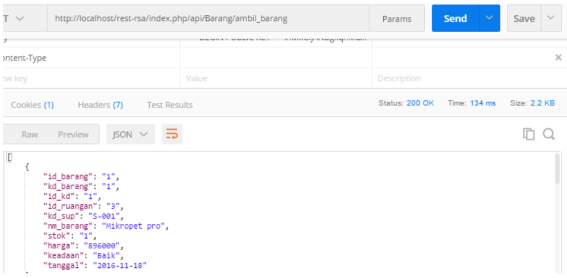


```
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzUxMiJ9.eyJrZF9wZWdhd2FpIjoiUC0wMDkiLCJ1c2VybmFtZSI6ImFyaWZpbiIsImlhdCI
6MTU0MjgwODUzOCwiZXhwIjoxNTQ1NDM4Mjg0fQ.TCNM52sU69pf3ellsxY57ozsokX8Kt8bzMlivX66pLMMRwwfM61Slr_hjsRV
zlSDR32VJfjD8pQvOZGHesGTUX1Glw41UZjsQrn-
L66oJkpwmU43bTjkJ2JvsMwn2KEj4jlvpB94jhVhzZHvjMKNtqmyIUIi3OJ7Tox174El6rDmIIw2FUrTR-
ySj25TrtQGHk6PiSEHxuB2PO6Z00Euca2cTocD5zitixOwK7s_SZiEGzSdL4wh1gMp2lUXlFR2ZmGvRzFkVW5_HDUnV2Akrbb6ib
jcd3oqS5oXDKyGXhDXc6aMPRaVCHV8rID6Dj9CpR2GZ-knZgLqEaW1VQXnJcWnzuDt1RtMBlzOxqwpenuM3GP8_sRXE1e_C-
w7Vy9O_AYHdZKg8ACxb6lvYpwmrfg88SkmTt8ycNhX2jr5UYl1Edb01qNglTatcEYbvsOB0s2BRQLEHPI_se91euN-
Xao71mhXVanAZ1baxsLX6JXPwHp8Wuaj7EiDeCpqITtOAoR_LfHDqHVsMagPwW5WOJR-733FqdYLx8VD_-1XzVOYjoVp3BN-
iPWHRDIh3nRr-eUhfYaiojM1R07uc-
3VYPg2z3gvKn0aQ1T1YohHhgUdtPNVWfCS09CaODMBkCRHoQyTqe9ZTRDBkbaC4x8qJfjCadAU53h3VAcoOys8C8k-----BEGIN
PUBLIC KEY-----MIICIjANBgkqhkiG9w0BAQEFAAOCAg8AMIICCgKCAgEAspMIHiDIG6R+gY6+Oc2E
Tprm/spav8nDyLJ1PNG3u5qWDg1isv3wl1DwaucJ27P+e/tZCR5s93HivYOLtqu73QtM9I1JYs4U+/mxFbJ
+4jWmUpP2A023aUqaXhv3CnQu6KeidYox/hTX+DcgM31b8bPfmJuBkF+LzGh7bkKRhoJ7JJEzr15BljulIV+vnDbuq7YSP6a3Ql
+fIo+XRQh7z0j4v/Uok7mcLM+G5ah1iYOltCL10xNU+NuuEYU4TxpjcNS6tETvOQXjEAKprzlUAqg
+WlsQJuzQo9KOaEjO1GC9+22Vmy2x8dGpMC0vgEr5jySkK3/E9WXNTuEHX/sY77YaIa2a+RkUHuXuoi/A+XtwuQ+B
+ey8qmMuCawfqI5C0eUWJ45vxZHKE6NsO/ce
pRUiHDFhRMvXQ94K2Ju6hmSg72yp7ijvglfhDWbb5mWYXuoUwCU6C7fA46JUpttO
X953aoFmUiGYvMNaR4tIcFn4Rt9WeY31KVL2AwqPxc3ZpX3h7h42r9/j/auH8T8uiNA+
+Rh0SrJlpqyYigFyfS61Ckre0yh5ye2g/JUVc7dBadpb5UPp37YMzvHMkMON
g/M4GRySyFGGq2O7u2DEH3Nf5qw31+UopswUaZeLMz6vFNmN6qR9fr8C08/DiToyEjDwQQ4vMPHtmcNz+LlubEUCAwEAAQ==
-----END PUBLIC KEY-----
```

Fig.5. Token.Txt in Restful Testing

Figure 5 is a token and public key generated from JWT RSA-512 that is implemented on RESTful WS. The size of the token produced is 1.58 KB.

Then the token and public key that was obtained from the POST results were used to GET request for API / items / take_items. The results of the GET process can be seen in Fig.6. JWT authentication on GET API / items / take_items with 134 ms time. The response generated is in the form of data items that exist in the database in JSON format.



http://localhost/rest-rsa/index.php/api/Barang/ambil_barang

Status: 200 OK Time: 134 ms Size: 2.2 KB

```
[
    {
        "id_barang": "1",
        "kd_barang": "1",
        "id_kd": "1",
        "id_ruangan": "3",
        "kd_sup": "S-001",
        "nm_barang": "Mikropet pro",
        "stok": "1",
        "harga": "896000",
        "keadaan": "Baik",
        "tanggal": "2016-11-18"
```

Fig.6. GET Request API/Items/get_items

B. Test Results

To see the performance of the JWT using the RSA-512 algorithm on RESTful WS, testing is done from the speed and size of the tokens generated, this test is done by testing 50 times of each generate token process and verifying tokens on RESTful WS using Postman tools. The experimental results of generating JWT RSA-512 on RESTful WS can be seen in Table 1.

Table 1. JWT RSA-512 Generate Comparison

No	Speed (*ms*)		Token Size (KB)	
	SOAP	***RESTful***	***SOAP***	***RESTful***
1	4216	2531	1,60	1,58
2	4166	2803	1,60	1,58
3	4948	3304	1,60	1,58
4	5075	3169	1,60	1,58
5	4205	3403	1,60	1,58
6	4286	3126	1,60	1,58
7	2047	3605	1,60	1,58
8	2142	3795	1,60	1,58
9	4748	2738	1,60	1,58
10	5886	3049	1,60	1,58
11	3666	3550	1,60	1,58
12	3640	2704	1,60	1,58
13	2831	2543	1,60	1,58
14	5837	3656	1,60	1,58
15	3880	2955	1,60	1,58
16	4076	3268	1,60	1,58
17	4164	2713	1,60	1,58
18	3019	3249	1,60	1,58
19	2906	2142	1,60	1,58
20	5737	2185	1,60	1,58
21	3935	2430	1,60	1,58
22	2963	2651	1,60	1,58
23	4538	3372	1,60	1,58
24	4340	2623	1,60	1,58
25	3553	2634	1,60	1,58
26	3460	2903	1,60	1,58
27	3981	2320	1,60	1,58
28	4943	2564	1,60	1,58
29	3204	2700	1,60	1,58
30	3446	2871	1,60	1,58
31	4184	2928	1,60	1,58
32	2696	3088	1,60	1,58

No	**Speed (*ms*)**		**Token Size (KB)**	
	SOAP	***RESTful***	***SOAP***	***RESTful***
33	2794	2972	1,60	1,58
34	3258	2363	1,60	1,58
35	4214	3442	1,60	1,58
36	3808	2578	1,60	1,58
37	3444	2833	1,60	1,58
38	4522	2887	1,60	1,58
39	3142	3467	1,60	1,58
40	2892	3031	1,60	1,58
41	3416	3392	1,60	1,58
42	4175	2629	1,60	1,58
43	4062	2681	1,60	1,58
44	3378	2767	1,60	1,58
45	4485	2697	1,60	1,58
46	2921	3109	1,60	1,58
47	4323	2337	1,60	1,58
48	3160	2303	1,60	1,58
49	4079	2451	1,60	1,58
50	3188	3073	1,60	1,58
Avg	**3839,58**	**2891,68**	**1,60**	**1,58**

Table 1 is the result of trying to generate tokens or request tokens on servers that have been done on SOAP and RESTful using postman tools. In this experiment, 50 experiments were conducted with the average speed of SOAP being 3839.58 ms, and the average speed at RESTful was 2891.68 ms. In terms of the size produced in SOAP that is with an average of 1.60 KB and in RESTful which is 1.58 KB. The average speed results in this experiment show that JWT RSA-512 is superior when implemented in RESTful. However, in terms of the size produced, SOAP slightly outperformed RESTful. The graph of the comparison of the speed of the generated token on SOAP and RESTful can be seen in Fig.7.

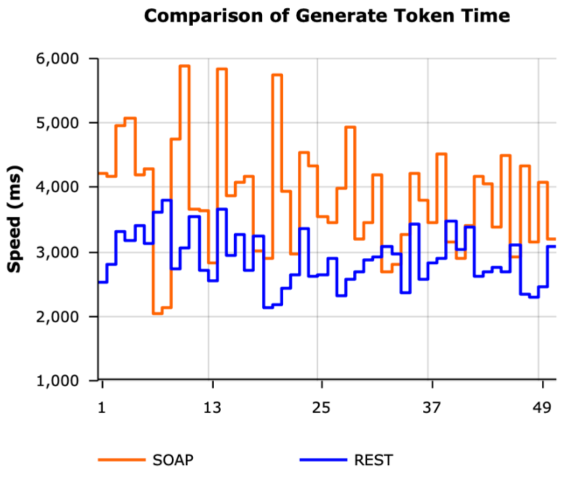


Fig.7. Comparison of Generate Token Time

In graph Fig.7, it can be seen that SOAP has a speed that tends to be slower than RESTful. This shows that JWT RSA-512's performance is more optimal when implemented on RESTful WS.

Then a performance comparison is performed on the authentication token process, or token validation and public key on SOAP and RESTful. The results of the experiment verify JWT RSA-512 can be seen in Table 2.

Table 2. JWT RSA-512 Authentication Comparison on SOAP and REST

No	**Speed (ms)**	
	SOAP	***RESTful***
1	133	135
2	125	131
3	157	120
4	172	148
5	177	122
6	116	128
7	125	112
8	117	125
9	144	130
10	134	135
11	168	116
12	167	137
13	170	120
14	164	119
15	188	135
16	106	130
17	114	117
18	120	124
19	121	100
20	166	127
21	139	131
22	124	139
23	117	125
24	120	116
25	128	130
26	164	126
27	122	140
28	149	119
29	125	116
30	133	114
31	141	120
32	147	129
33	123	119
34	127	126
35	147	117
36	146	112
37	135	100
38	110	123
39	161	120
40	154	120
41	164	112
42	135	153

No	Speed (ms)	
	SOAP	*RESTful*
43	138	109
44	155	112
45	123	142
46	164	125
47	162	119
48	138	137
49	123	134
50	137	120
Avg	**140,70**	**124,32**

Table 2 is the result of authentication token experiments by making requests using the generated tokens. In this experiment, 50 experiments were conducted with the average speed of SOAP, 140.70 ms, and the average speed at RESTful was 124.32 ms. The average speed in this experiment shows that RESTful is superior to SOAP. Graph comparison of authentication speed of tokens on SOAP and RESTful can be seen in Fig.8.

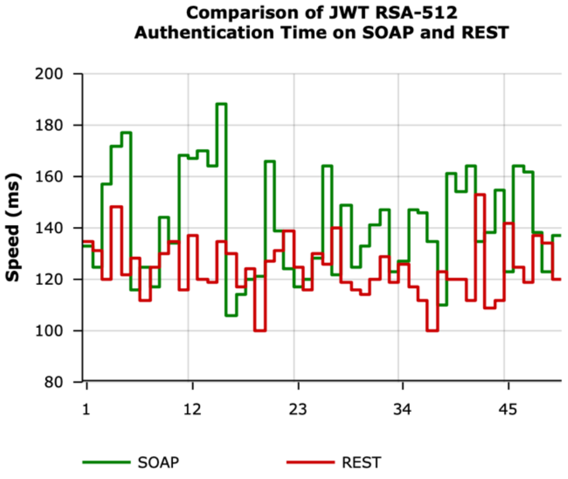


Fig.8. Comparison of JWT RSA-512 Authentication Time on SOAP and REST

In Fig.8, it can be seen that RESTful has speeds that tend to be faster than SOAP. This shows that JWT RSA-512's performance is more optimal when implemented in RESTful WS.

C. *Analysis*

The test results of JWT RSA-512 generate speed are 2891.68 ms. The test results of the JWT RSA-512 generated token has an average of 1.60 KB 1.58 KB. Then the JWT RSA-512 authentication test that has been done can be seen that the results of the speed in this test have an average yield of 124.32.

V. CONCLUSION

Based on the research that has been done, it can be concluded that the speed at the JWT RSA-512 token on the RESTful process tends to be slow, but quite stable. The resulting size of JWT RSA-512 token data is relatively small. Overall, JWT RSA-512's performance is very good when implemented in RESTful. For further research, we can test the JWT implementation using other asymmetric algorithms.

REFERENCES

[1] A. Rahmatulloh, R. Gunawan and I. Darmawan, "Web Services to Overcome Interoperability in Fingerprint-based Attendance System," in *2018 International Conference on Industrial Enterprise and System Engineering (IcoIESE 2018)*, Atlantis Press, 2019.

[2] H. Hamad, M. Saad and R. Abed, "Performance Evaluation of RESTful Web Services for Mobile Devices," *International Arab Journal of e-Technology,* Vols. Vol. 1,, no. No. 3, January 2010.

[3] R. Gunawan and A. Rahmatulloh, "Implementasi Web Service pada Sistem Host-To-Host Pembayaran Biaya Akademik," *Setrum: Sistem Kendali-Tenaga-Elektronika-Telekomunikasi-Komputer,* vol. 7, no. 2, pp. 320-329, 2019.

[4] OWASP, "OWASP Top 10 - 2017 The Ten Most Critical Web Application Security Risks," 2017. [Online]. Available: https://www.owasp.org/images/7/72/OWASP_Top_10-2017_%28en%29.pdf.pdf.

[5] M. I. Hussain and N. Dilber, "Restful web services security by using ASP.NET web API MVC based," *Journal of Independent Studies and Research – Computing,* vol. 12, no. 1, 2014.

[6] P. Sahoo, N. K. Janghel and D. Samanta, "Securing WEB API Based on Token Authentication," *International Journal on Advanced Electrical and Computer Engineering (IJAECE),* vol. 4, no. 2, 2017.

[7] X.-W. Huang, C.-Y. Hsieh, C. H. Wu and Y. C. Cheng, "A token-based user authentication mechanism for data exchange in RESTful API," *International Conference on Network-Based Information Systems,* pp. 601-606, 2015.

[8] A. Bhawiyuga, M. Data and A. Warda, "Architectural Design of Token-based Authentication of MQTT Protocol in Constrained IoT Device," *2017 11th International Conference on Telecommunication Systems Services and Applications (TSSA),* 2017.

[9] L. Xinhua, "The Design of Digital Campus Unified Identity Authentication System Based on Web Services," *Applied Mechanics and Materials,* pp. 2301-2304, 2013.

[10] I. I, P. M. R. Anand and V. Bhaskar, "Encrypted Token-based Authentication with Adapted SAML Technology for Cloud Web Services," *Journal of Network and Computer Applications 99,* 2017.

[11] P. F. Tanaem, D. Manongga and A. Iriani, "RESTFul Web Service Untuk Sistem Pencatatan Transaksi Studi Kasus PT. XYZ," *Jurnal Teknik Informatika dan Sistem Informasi,* vol. 2, no. 1, 2016.

[12] A. Rahmatulloh, H. Sulastri and R. Nugroho, "Keamanan RESTful Web Service Menggunakan JSON Web Token (JWT) HMAC SHA-512," *Jurnal Nasional Teknik Elektro dan Teknologi Informasi (JNTETI),* vol. 7, no. 2, 2018.

[13] RCBJ-ADMIN, "JWT Use Cases," 7 2017. [Online]. Available: http://rcbj.net/blog01/2017/07/14/jwt-use-cases/.

[14] V. Kumari, "Web Services Protocol: SOAP vs REST," *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET),* vol. 4, no. 5, 2015.

[15] M. A. Arianto, "Analisis dan Perancangan Representational State Transfer (REST) Web Service Sistem Informasi Akademik STT Terpadu Nurul Fikri Menggunakan YII Framework," *Jurnal Teknologi Terpadu,* vol. 2, no. 2, 2016.

[16] D. Oku, M. Yanagisawa and N. Togawa, "Scan-based Side-channel Attack against HMAC-SHA-256 Circuits Based on Isolating Bit-transition Groups Using Scan Signatures," *IPSJ Transactions on System LSI Design Methodology,* vol. 11, 2018.

[17] R. Gunawan and A. Rahmatulloh, "JSON Web Token (JWT) untuk Authentication pada Interoperabilitas Arsitektur berbasis RESTful Web Service," *JEPIN (Jurnal Edukasi dan Penelitian Informatika),* vol. 5, no. 1, pp. 74-79, 2019.