ISSN 2278-3091
Volume 9 No.2, March - April 2020
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse142922020.pdf
https://doi.org/10.30534/ijatcse/2020/142922020

iSSON- Intelligent Single Sign on Key Authentication Strategy using Powerful JWT Norms

Remya Chandran[1], Dr.A.Sasi Kumar[2]
[1]Ph.D Research Scholar, Department of Information Technology, School of Computing Sciences, Vels Institute of Science, Technology and Advanced Studies (VISTAS), Pallavaram, Chennai, Tamil Nadu, India.
[2]Professor, Department of Information Technology, School of Computing Sciences, Vels Institute of Science, Technology and Advanced Studies (VISTAS), Pallavaram, Chennai, Tamil Nadu, India.
[1]nivedika@gmail.com, [2]askmca@yahoo.com.

ABSTRACT

Now-a-days smart gadgets are the leading usage and communication mode accumulated by all people around globe. This drastic development leads many security issues and privacy-mismatches, in which all the commercial business organizations felt strange affections over their business goals. The major threat to deal with this kind of affection is based on authentication, whereas the authentication attack causes severe damage to the commercial or non-commercial records over the server, which automatically leads the system so poor and causes the performance down. There are many schemes available today, to provide security oriented authentication features and enable multiple clients around the globe to use privacy aware server and so on, however the practical problems are different, which causes the security issues over different levels like authentication, access control and many more. So, that to avoid this kind of security and privacy issues, an intelligent security-aware algorithm is required to resolve above quoted issues. In this proposed system, a new kind of key authentication strategy is introduced to provide higher end security norms to the gadgets and the associated servers by enabling privacy aware schemes, in which it is termed as Intelligent Single Sign On (iSSON) Scheme. This single sign on scheme allows the user to be sign in to the client system at once, the next time onwards the key enabling protocol creates a dynamic encrypted key to process the data and provides authentication for further with dynamically generated crypto keys. This kind of key generations are handled by Java-Web Token (J-WT). This proposed approach of Intelligent Single Sign On scheme is associated with Java-Web Token strategy to provide ultimate security features to the user/business environments. The association of Intelligent Single Sign On Scheme and Java Web Token Strategy is collectively called as "iSSON-JWT". This proposed system of iSSON-JWT provides users to attain high level of privacy along with reasonable costing nature and the practical implementation possibilities are really easy compare to the classical single sign on schemes in era. This proposed approach of iSSON-JWT give assurance to privacy enabling nature and cloud service robustness with practical proofs and it will be explained in detail over further summaries.

Key words: iSSON, Intelligent Single Sign On, Java Web Token, JWT, Cloud Service System

1.INTRODUCTION

With the prominence of cell phones, for example, Android-Smartphones and MAC gadgets, increasingly customary business exchanges are being directed online all day, paying little mind to their physical areas for whatever length of time that they have Internet Associations [1][2][3][4]. While this bears us numerous advantageous and associated idea of the wireless connectivities likewise uncovered the clients and framework to a wide scope of security threats [5][6][7]. A model mobile cloud-enabled Service design is appeared in the following figure, Fig.1. Right now, cell phones can get to cloud benefits in two modes: either through wireless network arrangement or through cloud service-providers. All the mobile clients expects many security norms to prevent their data around web medium by means of advanced security features such as crypto algorithms, machine-learning principles and so on. The circulated areas of the service handlers make it advantageous for client to access different services. Be that as it may, the plan of conveyed validation convention to guarantee secure correspondence while giving low computational overheads to these mobile clients is basic. As of late, a huge number of authentication-validations rule for single-server condition were introduced [6][7][8]. To oblige the wide extending furthermore, expanding requests for more extravagant and intuitive client service associations, many MultiServer platforms are sent in real-time applications. Anyway, existing login-verification laws intended for single-server platform, which are not appropriate for such MultiServer conditions, so, that a client needs to enroll in each and every server and recalls all credentials for the wide range of cloud-servers. Single Sign On authentication is a real-time deployment, since it permits a client with a solitary accreditation to get to various servers and allow accessing its features; therefore, this offer ascends to MultiServer Key-Authentication (MSKA) procedures. In regular single sign on plans, many open service architectures has been broadly developed by numerous analysts and some Internet specialist organizations, for example, Yahoo , Google and so on with more than 50k sites allegedly utilizing open source services as their key-authentication plot.

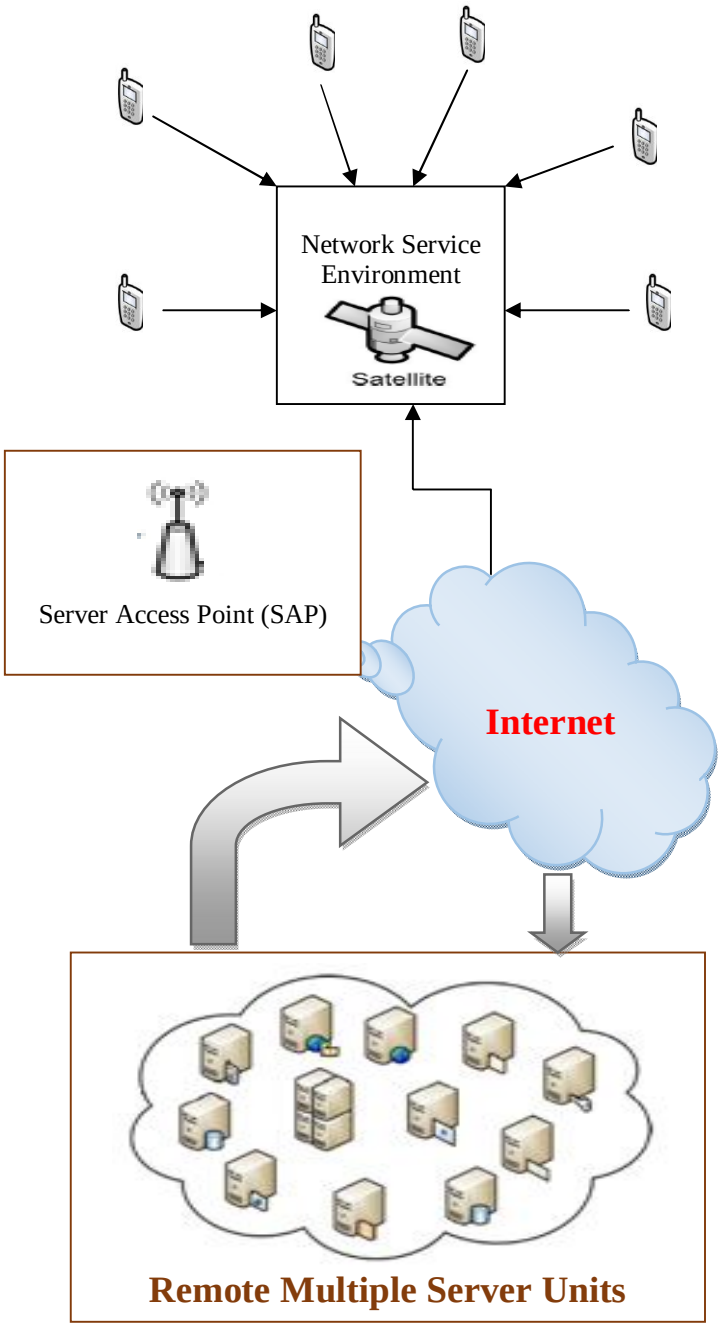

Figure 1: Remote Cloud MultiServer Architecture

In Open Service architectures, three parties are engaged with the key-authentication verification procedure. Moreover, the Open-source requirements firmly suggest the utilization of Secure Socket Layer ('SSL') organize association for all message transmissions. Since SSL method depends on Rivest Shamir & Adleman 'RSA' open key crypto nature, SSL execution requires overwhelming calculation costs. Such expenses may not be sensible for arrangement on a many wireless gadgets (for example at the point when service demands from device clients are considered) [9][10][11][12]. The 256'bit classical elliptic crypto methodology gives proportional degree of security as in a RSA'-open key crypto logic [11][12][13]. In this manner, elliptic crypto system is more competent for clients when contrasted with 'RSA'. The classical prerequisites of MultiServer key-authentication validations are client identity less authentications, Session-Key security and shared authentication validation. Many past works have additionally proposed the requirement for repudiation and re-enrollment highlights to be fused in MultiServer key-authentication laws because of the genuine and static verification token being lost/traded-off also, in this way utilized by an attacker to imitate the client [11][12][15]. As well as various MultiServer key-authentication rules have been proposed in the past systems, a large portion of these laws do not robust and also did not have have all the important security highlights. The procedure of a secure and robust MultiServer key-authentication convention supporting both denial and un-discernibility stays an exploration risk [12][13]. Existing MultiServer key-authentication conventions can be extensively arranged into tri-party based and dual-party based laws, refer Table: 1. In mode-1, an authorization community, a server and a client are associated with the key-authentication procedure and conventions in Mode-2 as it were comprise of a server 'S' and a client 'C' in the key-authentication process. In Mode-1, no MultiServer key-authentication convention with the exemption of Odelu'et-al. gives all important security highlights. We additionally see that most of the current Mode-2 MultiServer key-authentication conventions neglect to give all fundamental security highlights. This is the hole we look to address right now, we present a proficient protection and safeguarding-key based single sign on authentication convention for a MultiServer platform, which supports all fundamental security highlights.

Table 1: Existing System Protocols and Nature

Approach	Year	Protocol Used
MultiServer Key Authentication [10]	2008	Hash Key Function
Robust Biometric Authentication [15]	2013	Biometric Functions
Biometric and Cryptanalysis Improvement [14]	2012	Crypto Implementation with Biometric
Authentication Flaw [10]	2013	Dynamic Authentication
SAML Single Sign-On Service [11]	2017	SSOS
Privacy Aware Authentication [9]	2015	Privacy Hash Function

2. SYSTEM ANALYSIS

A. Existing System Summary

In past systems, many analysts work on Single Sign On scheme with multiple crypto service natures, but all are strucked up with certain limitations such as poor connectivity range, limited speed over multiple connections at simultaneous time and implementation complexity is really high because of its critical design nature. The past implementations highly believe MultiServer Authentication Key logics [1][2][3][10], in which it uses auto certificated open key nature to support clients over globe. But, this kind of MultiServer Authentication Key schemes are facing trouble over simultaneous attacks raised by intruders by using several sniffing tools, which is easily available over web markets [11][12]. This is also illustrated in multiple past research summaries such as [11][12][13]. Along with this past summaries [11][12][14] quoted this MultiServer Authentication Key scheme is not sufficient enough to protect the intruders trial over authentication attacks because of its periodical updations. The periodic updations are happened over some point of intervals, which is also easily judge able and the intruders can easily accumulate the created session keys by using sniffing techniques. Once the authentication key is gathered by the intruders, they can easily navigate into the respective user's portal and make modifications as well as access all the features according to their destructive needs. The existing protocols highly believe elliptic crypto logic to make secure the data while transmission [14][15]. However, the elliptical crypto protocols failed to achieve or guarantee security regarding user login-credentials, which won't consider the login credentials as the attribute to the respective user identity. The major existing protocols failed to support this kind of advanced security norms and the single sign on feature over past system believe two security norms such as MultiServer Authentication Key logics and Elliptical Crypto Service nature, but both of these schemes are failed to support client with ultimate privacy levels and protect their data with vulnerabilities or intruders. So, that a new scheme is required to resolve the problems quoted above.

B. Proposed System Summary

In the proposed system, our main focus is on enabling the security to data which is transacted between entities. For that a new authentication security mechanism is enabled and in which it provides advanced key cryptographic features along with Single Sign On logic. The proposed algorithm called Intelligent Single Sign On (iSSON) is in hands with powerful key generation policy called Java-Web Token (JWT) to provide the ultimate level of security to its users. By using this proposed mechanism, the user can make authentication into the system once at a time, after that the iSSON-JWT mechanism creates a unique and dynamic encrypted token, which is only shared with multiple servers for further needs. By using this approach, the user can prevent their identity in an efficient manner over cloud environments [1][2][10][11]. The following figure illustrates the work flow of the proposed system and further the Table-2 illustrates the parameter logics of the proposed system.

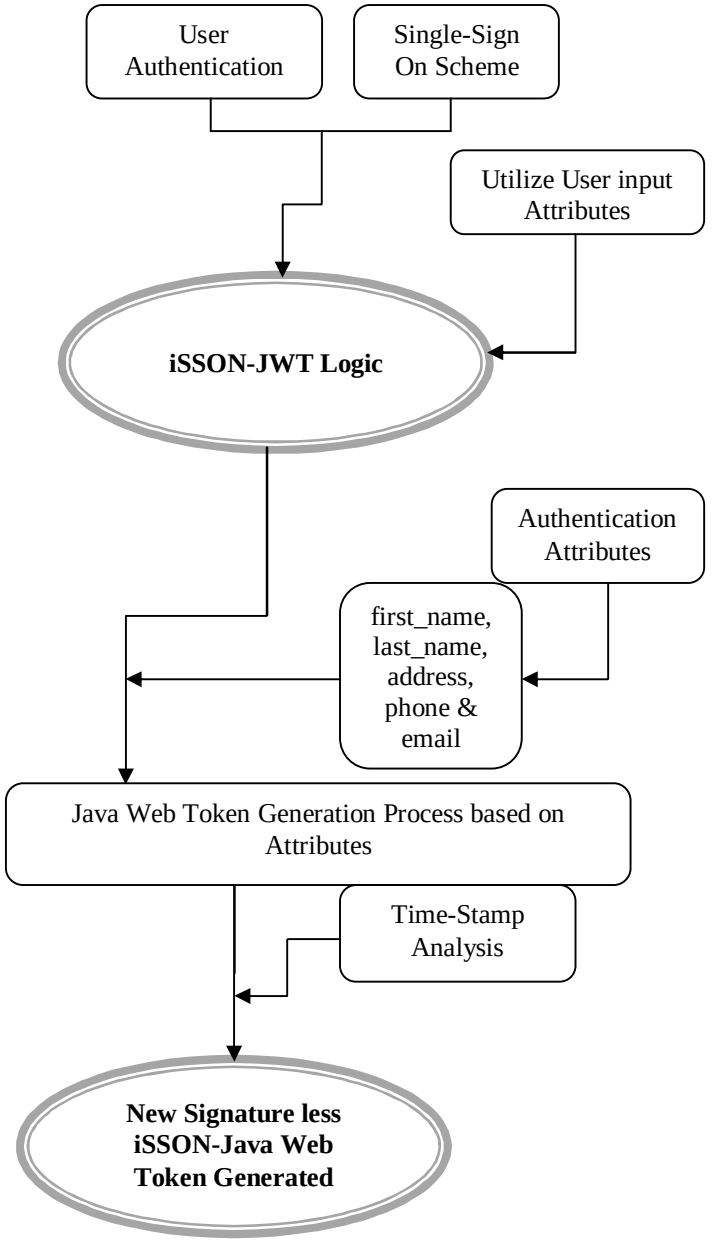

Figure 2: Work Flow Model of Proposed System

The above figure depicts that the work flow model and in which it starts from Authentication level along with Single Sign On strategy. The initial stage of work starts with the requirement of user input attributes and that will be input to the proposed algorithm Intelligent Single Sign On Java-Web Token (iSSON-JWT), in that the authentication attributes are the major factor to deal with. The authentication factors such as first-name, last-name, address, phone number and mail-id. Based on these input attributes the proposed logic JW Token generates a randomized dynamic token with proper timestamp as a consideration as well. This will be the powerful authentication norm and it is clearly illustrated with practical proofs over results and discussion section.

Table 1: iSSON-Java Web Token Single-Sign On Parametric-Symbols

Parametric-Symbols	Summary
RCens	Registration-Centers
U_i	User from 0 to n-1
IDen	Identity collected from user
GT	Generated Token
$H_{(0\text{-}n\text{-}1)}$	Hash Function for Crypto Laws
$DH_{(n\text{-}1,0)}$	Dehashing Function for Crypto Laws
$CServers^{(0 \text{ to } n\text{-}1)}$	Multiple Cloud Servers
C^p	Cost Level for implementations

3.SYSTEM IMPLEMENTATION

A. Intelligent Single-Sign On Tokenless Key Generation Procedure using Java Web Token Principles

The process of Single Sign On usually follows three procedures such as: Registration/Authorization, Key Generation and Authentication. These three process are individually handled by different algorithms like that the first module registration is handled by trust enabled services, the key generation process is handled by crypto mechanisms and the final authentication is handled by service providers or server side scripting units. These three modules are explained in detail below.

(i) Registration Phase:

The registration module in the proposed system of Intelligent Single Sign On procedure is fully handled by trust enabled mechanisms and it is the base for the authentication principles and the further stages of precedence. In this stage, the user has to enter their identities into the system such as first name, last name, username, password, mail-id, mobile number and so on. The proposed algorithm in association with Registration-Token Generation (RTG) algorithm to generate the dynamic web token to perform efficient user authorization in web medium. The iSSON-JWT enables the rich integrated support with this module, which is explained in detail below with proper algorithm and respective pseudo code.

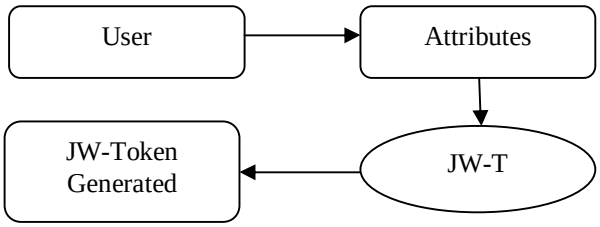

Figure 3: JW-T Token Generation Model

Algorithm: Registration-Token Generation (RTG)

Input: User Attributes
Output: Encrypted Web Token
Step-1: Collect User attributes.
Step-2: Attributes such as first name, last name, mail-id, mobile number, username and password are collected.
Step-3: Establishing the connection bridge between client end and the server.
Step-4: Enable the Javascript procedures by using JSON law.
Step-5: Get the inserted attribute details from Step-2.
Step-6: Validate the input.
Step-7: Check the retreived data values length is less than 1 or greater than 1.
Step-8: If it is greater than 1, generate the sign On function by calling JSON.stringify() and jwt.sign() function.
Step-9: Java Web Token Generated with the base of Step-4 to 8.
Step-10: Assign the Generated Java Web Token with the header (Ex. auth_token) for further verifications.

(ii) Key Generation:

The Key Generation process is typical in many scenarios of data security, but in case of single sign on logics irt is more complex compare to others, generally a key is used to handle the secure transaction between users in MultiServer environment to avoid interruptions from attackers or intruders. In the proposed system, advanced key generation logic is followed as similar to the classical techniques but the approach is different in the proposed system and for more detailos refers the above mentioned algorithm RTG.

Pseudocode: Registration-Token Generation (RTG)

```
NewUser = { first_name: req.body.first_name, last_name:
req.body.last_name, email: req.body.email, password:
req.body.password, address: req.body.address, phone:
req.body.phone }
const { valid, errors } = await validateSignupData(newUser)
if (!valid) return res.status(400).json(errors);
   let pass = req.body.password;
   const salt = await bcrypt.genSalt(10);
   const hashpassword = await bcrypt.hash(pass, salt)
   con.query("SELECT * FROM users WHERE email='" +
req.body.email + "'", function (err, result) {
    resultOut = JSON.stringify(result);
      if (result.length > 0) {
return res.status(400).json({ error: "Email is already in use" })
       } else {
sqlQuery = "INSERT INTO users
(first_name,last_name,email,password,address,phone)" +
```

```
"VALUES ('" + req.body.first_name + "','" +
req.body.last_name + "','" +req.body.email + "','" +
hashpassword + "','" + req.body.address + "','" +
req.body.phone + "')";
        con.query(sqlQuery, function (err, result) {
          if (err) {
            return res.status(500).json({ general: "something
went wrong please try again" })
          }
          resultOut = JSON.stringify(result);
          if (data.insertId.length != 0) {
tokenData = { user_id: data.insertId, name:
req.body.first_name, email: req.body.email, address:
req.body.address, timestamb: Math.floor(new Date() / 1000) }
data = JSON.stringify(tokenData)
const token = jwt.sign(tokenData, TOKEN_SECRET);
res.header('auth_token', token).send({code:200,token:token });
```

(iii) Authentication:

The authentication module allows the user to prove their identity into system and login for further precedence. In this proposed system, the authentication phase is fully handled by Intelligent Single Sign On principles. By using this approach user can login into the system with username and password one time and the further authentications are verified by using generated token and which is maintained into session for next usage of authentication into different server. So, that the security concern is highly preserved with this approach instead of login into multiple times over the server.

Algorithm: iSSON-Authentication

Input: User Credentials
Output: Boolean Result (True/False)

Step-1: Collect user credentials such as username and password.
Step-2: Decode the user attributes based on the given credentials.
Step-3: The decodes attributes over Step-2 are id, name, mail, address and respective timestamp.
Step-4: Assign the token data ionto JSON.stringify() function for validation.
Step-5: Pass these content into Java Web Token procedures with secret key as well, such as: jwt.sign(tokenData, TOKEN_SECRET).
Step-6: Check the decoded token is matched with the present user login state or not.
Step-7: if matched with the present user login, then allow the user to proceed further.
Step-8: Check the present timestamp and if it matches then the result state is marked as 1, otherwise the marked state is mentioned as 0
Step-9: Authentication Validation Succeeded.

The following details will clearly explain the Pseudocode used to validate the authentication token in real-time manner by means of the above quoted algorithm logics.

Pseudocode: iSSON-Authentication

```
const newUser = { email: req.body.email, password:
req.body.password }
const { valid, errors } = validateLoginData(newUser)
if (!valid) return res.status(400).json(errors);
con.query("SELECT * FROM users WHERE email='" +
req.body.email + "'", function (err, result) {
    resultOut = JSON.stringify(result);
    if (result.length === 0) {
return res.status(403).json({ general: "worng credentials
please try again" })
    } else {
compare(req.body.password, result[0].password, (err, match)
=> { if (match) {
tokenData = { user_id: result[0].id, name: result[0].first_name,
email: result[0].email, address: result[0].address, timestamb:
Math.floor(new Date() / 1000)}
data = JSON.stringify(tokenData)
const token = jwt.sign(tokenData, TOKEN_SECRET);
res.header('auth_token', token).send({code:200,token:token });
          } else {
return res.status(403).json({ general: "worng credentials
please try again" })
          }
if (err) {
return res.status(500).json({ general: "something went wrong
please try again" })
}
```

---

4. RELATED WORKS

This segment brief audits a few MultiServer authentication-Key-verification plans proposed in the paper. The structure of MultiServewr Key-Authentication conventions has not been a smooth excursion. For instance, in 2013, Liao and Hsiao proposed a matching based MultiServewr Key-Authentication convention utilizing self-certified open keys for mobile-customers. In any case, in 2014, Hsieh'and'Leu [14] exhibited that Liao' and 'Hsiao's convention [13] is defenseless to following attacks. Furthermore, it is called attention to that the convention is inefficient as each help server needs to refresh its identity table occasionally. Hsieh'and'Leu [14] then introduced an improved convention, which appropriates the static enlistment token with the dynamic hidden token. We see that the conventions of Liao' and 'Hsiao [13] and Hsieh' and 'Leu [14] utilize just transitory mystery keys (transient insider facts) in their authentication-Key-verification messages and meeting key scheme, which can be misused to encourage pantomime and estimated attacks [1][2][3][14]. Han' and 'Zhu [15] proposed an Elliptical cryptography-based MultiServer Key-Authentication convention, which was in this manner

demonstrated to be helpless against the security attacks [13]. Islam' [12] then proposed an security attack-free MultiServer Key-Authentication convention utilizing bilinear pairings. Be that as it may, Islams' convention neglects to guarantee the protection of client certifications as the identity is sent in plain content. We comment that greater part of the conventions in mode-2 receive a good/bad rundown to repudiate/grant clients' access benefits, without tending to the issue of denying a servers' access benefits. To have the option to renounce a server's access benefits, the server manipulation would need to advise every client and server [27].

In 2015, Tseng'et-al. built a rundown free MultiServewr Key-Authentication convention utilizing bilinear pairings. Be that as it may, their convention doesn't guarantee the protection of client accreditation since the client's character is sent in plain content. The convention likewise conveys the static validation tokens. As it were, Tseng'et-al's. convention doesn't bolster the fundamental security includes recently talked about. Right now, intend to structure a novel rundown free MultiServewr Key-Authentication convention utilizing elliptical curve crypto nature, which offers all notable security functionalities.

Table 2: Comparative Study

Branca Technique Provision	**Proposed JWT Provision**
Processing level is high, due to the number of iterations the Branca logic performs, which automatically leads the time consumption problem over proceedings [16].	The proposed JWT uses scalable authentication key states, which creates a token with pretty scalable norms, which is clearly illustrated over the proposed system summary of this paper over earlier section.
Traditional Branca based authentication strategy makes use of session and cookie to process the authentication data over server environment, in which this solution is difficult to process and due to this problem only scalability issues occur over the server [11][16].	Java Web Token procedures are using a stateless solution for key authentication and stateless applications, which are pretty easy to processing and easy to scale for further uses.
The total processing of Branca is still in a processing stage not yet concluded by the popular organizations, so that the resulting level we can treat it as a probabilistic based only [10][16].	Proposed JWT norms are popular and used by many OAuth Service Providers like Google, Facebook and so on. As well as it is very easy to verify jwt token and more trustworthy than cookie and sessions.
Data and Authentication Key or token tampering is complex and difficult to make while processing Branca logic in implementation level. As well as it is not scalable for sharing relevant data in different sizes [12][16].	Proposed JWT logic is used for authentication and they can also be used for sharing information, most of jwt are signed using a public key and a private key, therefore, it is very difficult to tamper with these token. JWT has three parts: head, body and signature, each separated by “dot(.)”.
Past hashing systems associated with Branca logic in SSO using regular Cipher logics to encode the authentication data over processing and decode that by using the same way [9][10][11].	The JWT authentication norms uses hashing in advanced crypto standards and the procedures are clearly explained in detail over Intelligent Single-Sign On Tokenless Key Generation Procedure using Java Web Token Principles, which is mentioned in this paper over Section.III

5.RESULTS AND DISCUSSION

The proposed resulting outcomes are illustrated clearly in the following summary. The following figure, Figure.4 shows the authentication page of the proposed system.

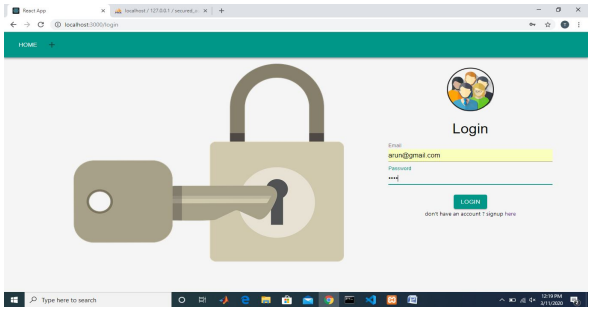

Figure 4: Authentication Page

The following figure Figure.5 illustrates the backend processing view of the user registration details of the proposed system.

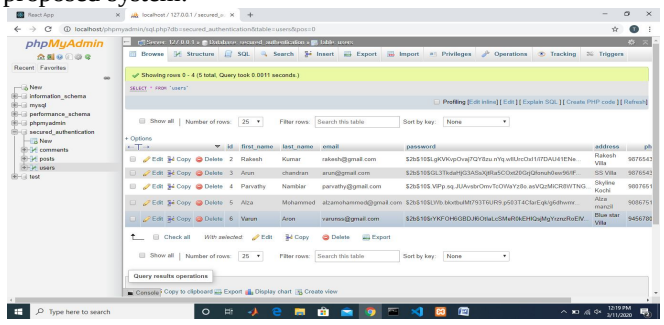

Figure 5: Registration Details

The following figure Figure.6 shows that the Post Submission portal back end processing view based on iSSON-Java Web Token Series of the proposed system.

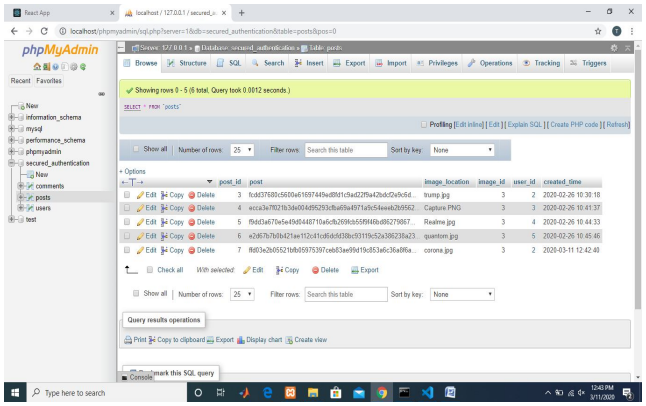

Figure 6: Post Submission Backend Processing View

The following figure Figure.7 illustrates the Posted Data and respective token details view of the proposed system.

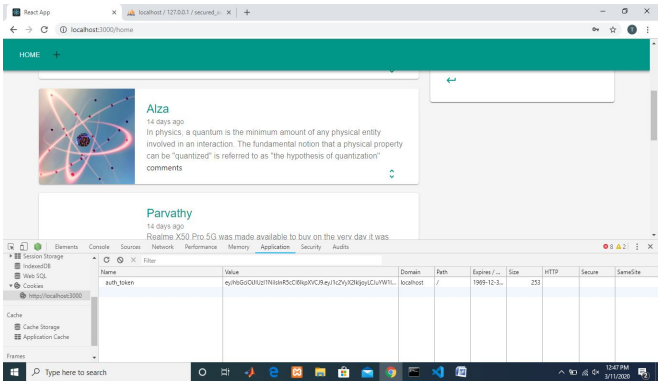

Figure 7 : Posted Data and Respective Token View

The following figure Figure.8 illustrates the Comment Section table view of the proposed system, in this we can easily get to know that the authentication key is not stored in server.

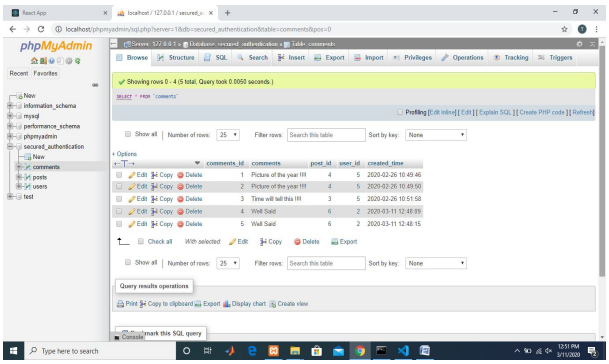

Figure 8: Comment Section Database Maintenance View

6. CONCLUSION AND FUTURE SCOPE

In this paper, an intelligent Single Sign On based Java Web-Token (iSSON-JWT) Strategy is implemented, which is more suitable to the MultiServer Authentication strategies and provides improved security and privacy schemes compare to the past implemented techniques such as Branca and so on. Many past server side schemes are available for Single-Sign On scheme in literature', but all are suffered under certain lackings such as uni-directional authentication norms, insufficient security parameters, poor performance and so on. In the proposed iSSON-JWT based application provides an efficient way to resolve all the above mentioned problems clearly under several cases such as speed, security and reliability. The proposed system follows the encryption strategy of Java Web Token principles, which is operating under the principles of powerful JavaScript based Advanced Encryption Standard (AES-JS). The proposed system provides bi-directional authentication norms and high-security enabled credential data maintenance port, which illustrates the strength of the proposed Intelligent Single Sign On strategy stronger than past schemes.

In further the proposed work can be enhanced by means of adding deduplication facilities to avoid unwanted server filling and the storage improvements automatically improves the performance of the entire system.

REFERENCES

[1] S.'Abolfazli, 'Z.'Sanaei, 'E.'Ahmed, 'A.'Gani,' and R.'Buyya, `Cloud based augmentation for mobile devices 'Motivation, 'taxonomies,'and open challenges, "IEEE Commun.' Surveys Tuts.,'vol.'16,'no.'1,'pp.'337368,'1st Quart.,'2014.

[2] D.Wang,'D.'He,'P.Wang,' and C.-H.'Chu,'``Anonymous two-factor authentication in distributed systems 'Certain goals are beyond attainment, "IEEE Trans. 'Depend. 'Sec.'Comput.,'vol.'12,'no.'4,'pp.'428442,.'2015. https://doi.org/10.1109/TDSC.2014.2355850

[3] K.-K.'R.'Choo,'High tech criminal threats to the national information infrastructure, "Inf. 'Secur. 'Tech. 'Rep., 'vol. '15,'no.'3,'pp.'104111,'2010. https://doi.org/10.1016/j.istr.2009.09.001

[4] K.-K.'R.'Choo,'``The cyber threat landscape 'Challenges and future research directions,' 'Comput. 'Secur., 'vol.'30, 'no. '8, 'pp.'719731,'2011. https://doi.org/10.1016/j.cose.2011.08.004

[5] N.'H.'A.'Rahman and K.-K.'R.'Choo,'``A survey of information security incident handling in the cloud, "Comput.'Secur.,'vol.'49,'pp.'4569,'Mar.'2015. https://doi.org/10.1016/j.cose.2014.11.006

[6] K.-K.'R.'Choo, 'Secure Key Establishment, 'vol.'41. 'Boston,'MA,'USA'Springer,'2008. https://doi.org/10.1007/978-0-387-87969-7

[7] K.-K.'R.'Choo,'`Organised crime groups in cyberspace 'A typology,' 'Trends Org.'Crime,'vol.'11,'no.'3,'pp.'270295,'2008.

[8] M.'Ge,'K.-K.'R.'Choo,'H.'Wu,'and Y.'Yu,'``Survey on key revocation mechanisms in wireless sensor networks, "J.'Netw.'Comput.'Appl.,'vol.'63,'pp.'2438,'Mar.'2016.
[9] J.-L.'Tsai and N.-W.'Lo,'``A privacy-aware authentication scheme for distributed mobile cloud computing services, "IEEE Syst.'J.,'vol.'9,'no.'3,'pp.'805815,'Sep.'2015.
https://doi.org/10.1109/JSYST.2014.2322973
[10] A.'Armando, 'R.'Carbone, 'L.'Compagna, 'J.'Cuéllar, 'G.'Pellegrino ,'and A.'Sorniotti,'``An authentication aw in browser-based single sign-on protocols 'Impact and remediations,"Comput.'Secur.,'vol.'33,'pp.'4158,'Mar.'2013.
https://doi.org/10.1016/j.cose.2012.08.007
[11] Google.'(2008).'SAML Single Sign-On (SSO) Service for Google Apps. 'Accessed 'Aug.'2017.
[12] OpenID Foundation. 'The OpenID User Interface Extension Best Practices for Identity Providers 2009.'Accessed 'Aug.'2017.'
[13] E.'Barker, 'W.'Barker, 'W.'Burr, 'W.'Polk, 'and M.'Smid,'` `Recommendation for key management part 1'General (revision 3),"NIST Special Publication, 'vol.'800, 'no.'57,'pp.'1147,'2012.
https://doi.org/10.6028/NIST.SP.800-57p1r3
[14] J.-L.'Tsai,' ``Effcient multi-server authentication scheme based on oneway hash function without verication table,"Comput.'Secur.,'vol.'27,'nos.'34,'pp.'115121,'2008.
https://doi.org/10.1016/j.cose.2008.04.001
[15] E.-J.'Yoon and K.-Y.'Yoo,' ``Robust biometrics-based multi-server authentication with key agreement scheme for smart cards on elliptic curve cryptosystem.
[16] Remya Chandran and A. Sasi Kumar, "An Efficient Keyless Signature and Improved Version of Merkle Signature Scheme-CMSS", International Journal of Engineering and Advanced Technology (IJEAT), Volume 8, Issue 5, June 2019.
[17] Remya Chandran, A.Sasi Kumar, "Keyless Signature Infrastructure Solution For Cloud Attack", International Journal of Engineering and Technology, Volume 7, Issue 2.33 (Special Issue 33), pp.513-514, 2018.
https://doi.org/10.14419/ijet.v7i2.33.14822