# A Novel Authentication Mechanism to Prevent Unauthorized Service Access for Mobile Device in Distributed Network

https://doi.org/10.3991/ijim.v12i8.8194

V. L. Pavani
Madanapalle Institute of Technology and Science, Madanapalle, A.P., India
pavaniveluru1@gmail.com

**Abstract—**The growth of distributed computer networks (DCN) is simple for the user to share information and computing capabilities with the host. User identification is an essential access control mechanism for the client-server networking architecture. The perception of single sign-on allows a legitimate user to access a different service provider on a DCN using a single session key. Recently, several user identification techniques being proposed for DCN. Unfortunately, the existing proposals cannot maintain user anonymity when the majority of probable attacks. In addition, the further time synchronization mechanism they use can result in widespread overhead costs. To overcome these shortcomings, we propose a novel authentication mechanism to prevent unauthorized service access for a mobile device in distributed networks. The mechanism implements methods to generate securely encrypted keys using RSA algorithm to validate the authentication of user login id and password. Later, it implements a secure session key generation using the DH algorithm which allows accessing the different services without repeating the authentication mechanism. An experimental evaluation was performed to measure execution time overhead for the registration process, an authentication process, session key generation process and Service Request Performance. The comparison results with existing authentication mechanism show an improvisation in all the measures.

**Keywords—**Authentication, Unauthorized Service Access, Mobile, Distributed Network.

## 1 Introduction

In a real-world application, mobile users can use mobile devices such as mobile phones with a single session key to access a number of services such as music downloads, e-mail reception or reply, product orders, or online payments. etc., from the different service provider in a distributed computer network (DCN). In a typical solution, users have to register with each "service provider" and maintain different ID/password pairs to access each service provider. However, if users need to keep a lot of confidential information, security issues can arise and network overhead can increase. Many research works are being proposed in [1], [2], [3] and [4] which provide

a user authentication protocol through session key establishment for the user anonymity in a DCNs. For those attacks that do not protect the secret session key of an attacker, an attacker can falsify the service provider by forging a legal session key.

Authentication is a horizontal requirement in many applications, platforms, and infrastructures [5], [6], [8], [15]. Distributed computer networks have a number of security issues addressed by user identification schemes and provide many improvements to handle attacks that provide "session key establishment", "user anonymity", based on the timestamp and do not provide efficient results at different time zone, or if there is a congested network environment with unstable latency, an additional time synchronization mechanism is required. A duplicate set of data occurs because all users in the organization must be set up for the service provider's application.

The "Single Sign-On (SSO)" model is a single task of user authentication and authorization that allows a user to access all the resources of the user's application [10], [11], [13]. It does not have to enter multiple passwords to gain access. The concept of SSO can also be applied to inter-organizational relationships, and users are free to explore related sites within the "bounds of trust". However, for many "SSO" means that each user of the service provider has only one user ID and password [25], [26]. For others, "SSO" means that each time a user logs in, the user is provided with a customized interface and a set of applications. Another preferred interpretation of the author is "SSO", which aims to provide end users with only one authentication attempt during a single work session. Because different service providers are under completely different administrative control, it is difficult to maintain a common SSO throughout the service. This makes it very difficult to log in once and gain authenticated access to multiple services.

In this paper, we propose a novel authentication mechanism to prevent unauthorized service access for a mobile device in distributed networks through a secure SSO login and dynamic service session key generation. The mechanism implements the methods to generate securely encrypted keys using RSA algorithm [9] to validate the authentication of user login id and password. Later, it implements a secure session key generation using the DH algorithm[17] which allows accessing the different services without repeating the authentication mechanism.

The purpose of this proposal is to solve mobile device service issues that are accessed from multiple service providers using limited resources in relation to the battery, bandwidth, and computational constraints. Most mobile applications access the information by performing SSO to registered sites and sharing authentication details with multiple service providers. However, in order to gain access, multiple applications must be logged in repeatedly to obtain authentication. This results in lost battery and increased computation and communication overhead for mobile devices. Based on the proposed content, we solve repeated login problems and weaken unauthorized access to the service.

The rest of the paper organized as follows. Section-2 discuss the background study, Section-3 presents the proposed novel authentication mechanism methodology, Section-4 provide the experiment and results in evaluation and section-5 present the conclusion of the proposal work.

## 2 Background Study

Significant improvements in emerging technologies now make the mobile more like a minicomputer, and the advanced technology on the phone makes it easy to access the versatile Internet. However, the security issues that users need to resolve must determine whether the user is legitimate, authenticate the service provider, and set the common session key appropriately to support the legitimate user's personal information. In the past, a user identification protocol that gives "session key establishment" and "user anonymity" for a DCN disguised as an attack without protection, these attacks can falsify legal session key that legitimate session key can deceive service providers[2], [3], [6], [12]. Later, several methods have been proposed that are vulnerable to identity disclosure attacks and suggest improvements to prevent such attacks. [19] [22], [23], [24].

Several approaches have been proposed that suggest mechanisms for identifying vulnerable public attacks and suggest improvements to prevent such attacks [4], [5], [7], [14], [16]**.** The idea for the SSO platform solves the problem by handling authentication to different software systems using one unique central account database and one login procedure I will. It handles access control for several related but independent applications. Users can get approval for all applications in the SSO trust environment by logging into a single system without prompting them to log in again when the platform changes [18]. The SSO proposal is a seamless connection of the authentication process that is performed at different levels. Passing the signed token during the network access phase offers the needed functionality to bootstrap the SSO system. The SSO specification created by Open Group relieves the burden of system administrators who need to control access to distributed systems, and users who need to remember multiple passwords to access multiple systems. It provides a unified mechanism for implementing business rules that govern user authentication and determine user access to applications and data.

The SSO solution [10], [11], [13], [25] can be a solution to maintain a large number of user IDs and passwords, but at the same time difficult to get all the application services, Implementing SSO decreases some security risks, but enhances other security risks. For instance, if a malicious user remained logged in and away from his computer, all authorized resources could be compromised. If it logs on at least several times, it can only log in to one system at a time, so the only individual resource is corrupted. With SSO, all applications can use a single central authentication service. This is a striking goal for hackers who can choose to perform a "denial of service attack".

S. D. Yalew et al. [1] presents the design of "TRUAPP", a software authentication service that ensures the reliability and integrity of apps running on mobile devices. It takes advantage of "ARM TrustZone hardware" security extensions to provide this assurance even in the event of an operating system corruption. It uses technologies such as "static watermarking", "dynamic watermarking" and "cryptographic hash" to verify the integrity of the application. This service was implemented on a hardware board that emulated a mobile device, which was used to perform the experimental evaluation of the service.

D. Davidson et al. [3] discusses security weaknesses in the interface between app code and web content through attacks, and then introduces defenses that can be

deployed without modifying the OS. The "WIREframe" allows both apps and embedded web content to define simple access policies to protect their resources. These policies recognize granular security principals like the original and control all interaction between the app and the web.

R. Peeters et al. [2] discuss weak security, excessive personal data collection for user profiling, user experience issues, and mobile authentication issues. Despite an interesting platform, mobile devices still do not have enough potential for authentication. It proposes "n-Auth", a definitive step that opens up the full potential of mobile devices for authentication by improving security and usability while respecting the privacy of users. We combine several secure, encryption technologies with secure HCI design principles to focus on getting a better user experience.

J. Costa et al. [6] describes a lightweight two-factor authentication system in which a legitimate user uses a mobile device to access a particular service. The "Two-Factor Authentication (TFA)" is becoming increasingly important for user security and identification. As cybercrime increases, businesses that run from financial institutions to retailers every year implement a TFA mechanism to ensure the trustworthiness of users within the system, reducing the risk of malicious users penetrating the system.

C. C. Chang et al. [10] presented an interesting "RSA-based SSO scheme" depend on a "one-way hash function" and a "random nonce" to solve the weakness of the timestamp and reduce the overhead of the system. It is very efficient in terms of calculation and communication costs. The parameters used here are the calculation cost and the communication cost. However, G. Wang et al. [11] Chang Lee's plan has shown that it is not safe to recover credentials without credentials and apply "impersonation attacks". The primary attack is "credential attack" that compromises the credentials privacy is a malicious service provider that has effectively communicated with a legitimate user two times, after recovering the user's credentials and then impersonating the user, To be accessed. Other attacks, such as impersonation attacks without compromising credentials, allow untrusted outsiders to freely benefit from network services by pretending to be legitimate users.

While many authenticated key agreement protocols are discussed in various articles [20], [21], [22], they provide mutual authentication and key exchange mechanisms. A major drawback is the high computational cost. The "RSA algorithm" is the generally utilized for "encryption" and "authentication algorithm" through the addition of two sets of numbers that make up a public key and one more set of private keys. Both the "public and private keys" are required for encryption or decryption, but only the private key owner needs to know about it. A new authentication mechanism based on the "RSA" and "Diffie-Hellman algorithms" for ensuring mobile user identification and efficient access to services is proposed in understanding the limitations and disadvantages of the authentication and privacy preservation schemes present in existing literature.

## 3 Proposed Authentication Mechanism

We propose here a novel authentication mechanism architecture as given in Fig.1. It normally refers to the anonymity state of an "individual's personal identity", or "personally identifiable information", being openly anonymous. It presents an SSO which enables the user to authenticate once in order to access many resources and will allow mobile users to use the "DH session key (*DSKey*)" to access service providers.

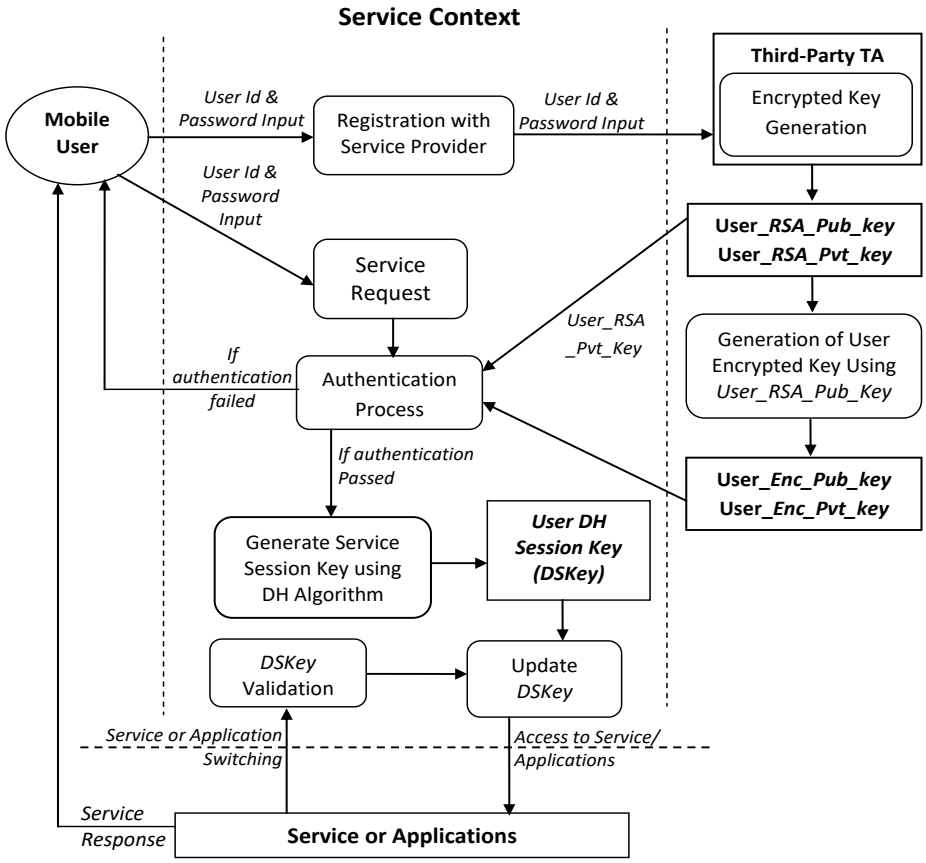



**Fig. 1.** Proposed Authentication Architecture

Authentication is the first step in proving user identity when it sees a system login referencing identity management. The confidential login data provided to the employee upon enrolment is compared with the data in the lookup table where the personalized information is stored and only the user is granted access to the service. In a "unidirectional identification scheme", an entity recognizes the other party by testing various confidential information. In addition, the common identification protocol allows two communication parties to authenticate each other. Therefore, there is a significant

security issue that requires that the "user identification scheme" be addressed to the user. Whether the user is legitimate or not, the service provider must be authenticated, the general session key has to be properly set up to access the various services, and the legitimate user's personal information must be retained.

The functionality of the mechanism can be divided into three modules. The first module ensures the user registration and authentication mechanism, the second module discuss the mechanism of service session key generation (*DSKey*), and the third module presents the mechanism of exchanging of *DSKey* for restricting the unauthorized Service Access.

### 3.1 User Registration and Authentication Mechanism

The process of user registration and authentication process initially defines a method for registration, where the user provides its user id and password as an input. The input user id and password will be submitted to "Third-Party Trusted Authority (TA)" to provide a user "RSA public key (*User_RSA_Pub_Key*)" and "private key (*User_RSA_Pvt_Key*)". Based on the generated *User_RSA_Pub_Key* the input user id and password get encrypted to create a user encrypted public key (*User_Enc_Pub_Key*) and (*User_Enc_Pub_Key*) private key. This generation mechanism is illustrated in the Fig.2.

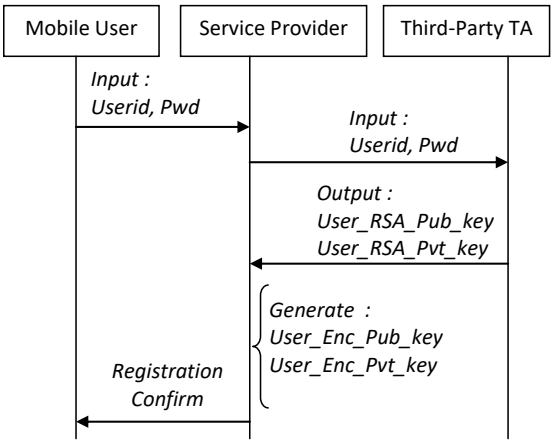



**Fig. 2.** Registration Process and Key Generation

On completion of registration process, the user tries to access the services through providing it login details. The authentication process evaluates the login details through the following Algorithm-1.

This authentication mechanism improves SSO solutions that support trusted third-party rights. Secured by a third party (TTA) The RSA public and private key is the primary key to generating user-encrypted public and private keys. This overcomes the traditional way of storing usernames and passwords in a simple symmetric encryption format that is susceptible to Brute Force attacks. The encrypted key is being traced, but

the TTA private key is required for the proposed method to open. This provides a strong association between the user and the TTA key to improve the SSO authentication mechanism.

There are a variety of ways that service providers implement SSO, but common industry standards use Identity Server and Web agent-based solutions. In traditional systems, multiple systems have independent authentication mechanisms, and each time a user needs to refer to each system differently each time they access the system, there is no unique interface. SSO for medium and large service provider systems can be a "single point of failure", if not appropriately designed. If the SSO system is down but the service provider is still present, users can lose access to all resources or applications protected by the SSO system and thus lose QoS.

**Algorithm-1:** User Authentication

***Input:*** UserId, *U,*
Password, *Pwd*,

***Output:*** Validation Status, $V_{status}$

***Method:*** **User_Authentication (*U*, *Pwd*)**

$V_{status}$ = *false;*

*//-- User Generated Encrypted keys*
Enc_pub_key = Read *User_Enc_pub_key ;*
Enc_pvt_key = Read *User_Enc_pvt_key ;*

*//-- RSA Key by Third-Party TA*
RSA_pvt_key = Read *User_RSA_pvt_key;*

*//-- Conversion the keys into BigInteger*
$Z_{pub} = \mathbb{Z}\,(Enc\_pub\_key)$;
$Z_{pvt} = \mathbb{Z}\,(Enc\_pvt\_key)$;

*//-- Get the key into Exponential factor*
$d = \mathbb{Z}\,(RSA\_pvt\_key)^{exp}$ ;

*//-- Get the key Modulus factor*
$n = mod\ (RSA\_pvt\_key)$ ;

*//-- Decryption of Keys*
$C_{pub} = d^n\ mod\ (Z_{pub})$;
$C_{pvt} = d^n\ mod\ (Z_{pvt})$;

*//-- Convert of Keys into String for validation*
E_Uid = $toString(C_{pub})$;
E_Pwd = $toString(C_{pvt})$;

If ( $E\_Uid == U$ ) and ($E\_Pwd$ = = Pwd)
$V_{status}$ = *true;*
End If

So, depending only on SSO authentication mechanism cannot assure the continuous access and even unauthorized access to services. If a user able to make a successful login through an authentication mechanism then a dynamic service session key (DSKey) generation will be initiated. It will provide a unique DSKey using Diffie-Hellman Algorithm which will be valid for a service or application session. It will update automatically with a new DSKey when user switching between the services or application. The complete mechanism of user session key generation is discussed in section 3.2.

### 3.2 User Service Session Key (DSKey) Generation

Since the internet is stateless, the software must verify all requests in the user's browser to determine if there is an authentication policy associated with the "resources" or "applications" that the user is trying to access. However, validation each time a user clicks on a diverse URL may result in additional overload and traffic congestion. Devices with limited resources such as mobile, PDAs, etc. can experience high resource losses. Therefore, it is very important to have an independent module that can provide the session key to the service without repeating the authentication process.

We suggest a mechanism to create user service session key as "*DSKey"*, using "Diffie Hellman algorithm". As Diffie Hellman public-key method uses "asymmetric key principles" for the distribution of "symmetric keys" to both parties in a communication network. Key sharing is an important aspect of the conventional algorithm and the entire safety is dependent on the distribution of key and it frequent updating. The generation of "DSKey mechanism" utilizes the public& private key of asymmetric key cryptography[29] of Diffie-Hellman as is presented in Algorithm-2.

**Algorithm-2:** User Service Session Key Generation

***Input:*** Secret first prime factor as, *P,*
Secret second prime factor as, *G, where G<P.*

***Output:*** Session Key as *DSKey*

***Method:*** **Generate_Service_Session_key (*P*, *G*)**

*//-- Generate a random number, R as the private key*
$R = Generate\_Random\_Number\ (100);$
*while (R > P) do*
R = R - P;
*End While*

*//-- Generate a Dynamic Session Key as DKey*
$DKey = G^R\ mod\ (P)$ *;*
$DSKey = msg(R, DKey);$

Based on the generated DSKey the authenticated user can access the services provided by service providers. In exchange for this key service provider implements a DSKey validation module which authorized the accessibility of the user. The mechanism of exchanging and validation of DSKey is presented in section 3.3.

### 3.3 Exchanging and validation of *DSKey* for Unauthorized Service Access

Exchanging of the key allows parties or service provider to enable information sharing in a distributed and multiple services. It can act as an authenticated code between two parties to ensure data sharing confidentiality and integrity. The generated "*DSKey"* is loaded for a session initiated for the service user is requesting for, and if the user is navigating in the same service context the same session key will remain as most traditional approaches do. But issues raise when the user moves out the current service context to another service, for example, user currently browsing for music download in site-1 and it switches to site-2 for different music, this change of context may ask for a re-login for the authentication for getting ensured of the identity of the other accessing users, but this can be time, cost and resource consuming process.

We suggest a novel mechanism for exchanging of session key between within service context and outside service context to reduce computation overhead and improvise QoS in accessing service with less delay. The validation of DSKey is depended on the service context switching as illustrated in Fig.3.

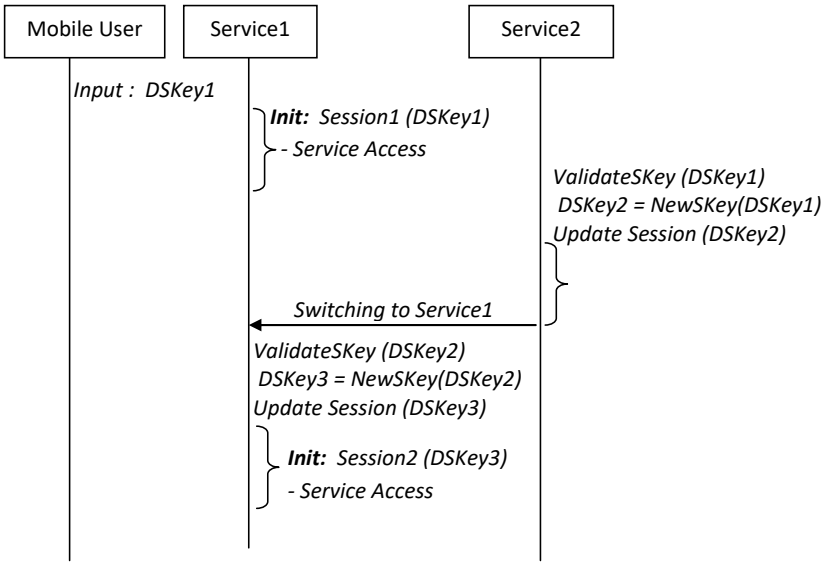



**Fig. 3.** An Illustration of DSKey Exchange Mechanism

As the exchanging of the cryptographic session key is generated through a Diffie–Hellman algorithm which allows two services context that have no prior knowledge of each other, so both jointly establish a new session key to identifies the unauthorized

access of the services. The Algorithm-3 describe the mechanism of DSKey validation and new session key process.

**Algorithm-3:** DSKey Validation and New DSKey Generation

*Input:* Service1 session key, *DSKey1,*
Secret first prime factor as, *P,*
Secret second prime factor as, *G, where G<P.*

*Output:* New Session Key as *DSKey2.*

*Method:* **ValidateDSKey(***DSKey1***)**

*Validation Status as,* $S_{status}$ *= false;*

*//-- Extract user as DH key, and session key*
$DKey = getDKey(DSKey1);$
*UKey = getUKey(*$DSKey1$*) ;*

*//-- Generate valid random number, V as the private key*
$V = Generate\_Random\_Number\ (100);$
*while (V > P) do*
V = V - P;
*End While*
*//-- Generate Validating Key as VKey*
$VKey = G^V\ mod\ (P)$ *;*

*//-- Create Comparison key*
$VSKey = (DKey)^V\ mod\ (P)$ *;*
$USKey = (VKey)^V\ mod\ (P)$ *;*

If (*VSKey == USKey* )
$S_{status}$ *= true;*
*DSKey2 =* ***Generate_Service_Session_key (P, G);***
End If

The goal of this dynamic session key generation and exchanging provides assurance that no unauthenticated user can pretend and established session key to access the service indirectly. The proposed authentication mechanism experiment evaluation and outcomes are discussed in the following section.

## 4 Experiment Evaluation

This section describes the experiment evaluation methodology and technologies utilized in the proposed mechanism. The design architecture consists of two main sides as, *User side* and *Server Side*.

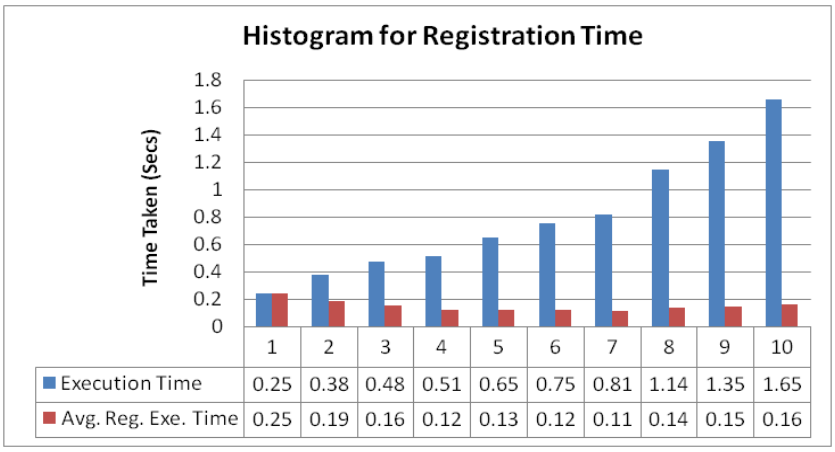
a) ***User Side Setup*:** A user is considered as a mobile node use performs the initial registration activities and later request for services through the authentication mechanism. A java based multithreaded nodes are created evaluates the multiple nodes requests.
b) ***Server Side Setup***: We run a registration server with Third-party TA using Apache web server. We also implement a separate module to evaluate the user authentication mechanism during login. The mechanism is implemented using the Java security API. It also implements the session key creation module in case of a successful login.

In order to prove the effectiveness of our approach, we tested the performance of the basic cryptographic algorithms proposed that we have used throughout our proposal as well as the main functions of our architecture. For the needs of our experiments, we used Nodes own built-in timer in the process module implemented through JavaScript [27].

### 4.1 Result Analysis

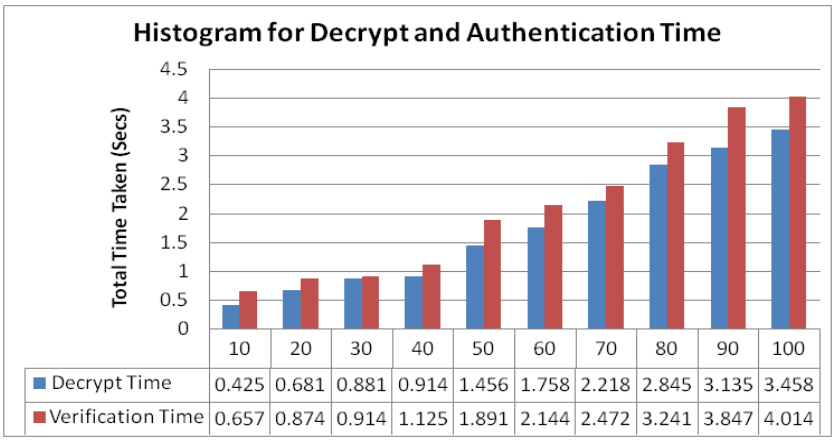
**Registration Execution Time.** We measured and analyze the time that it needed to complete the registration function on the server, with an increasing number of request from user 1 to 10. The processing time taken at a server was measured using Apache JMeter [28]. It measures the time taken by a server to execute the function code for each number request. Fig. 4 illustrate the result obtained.

**Authentication Execution Time.** We analysis here the measure of the execution of decryption of encrypted keys for user authentication mechanism to the verification of user login. To measure the efficiency of the process we run the iteration from 10 to 100 request. The Fig. 5 shows the total decryption time taken with increasing number authentication request, and Fig. 6 shows the complete time taken to do the verification process.



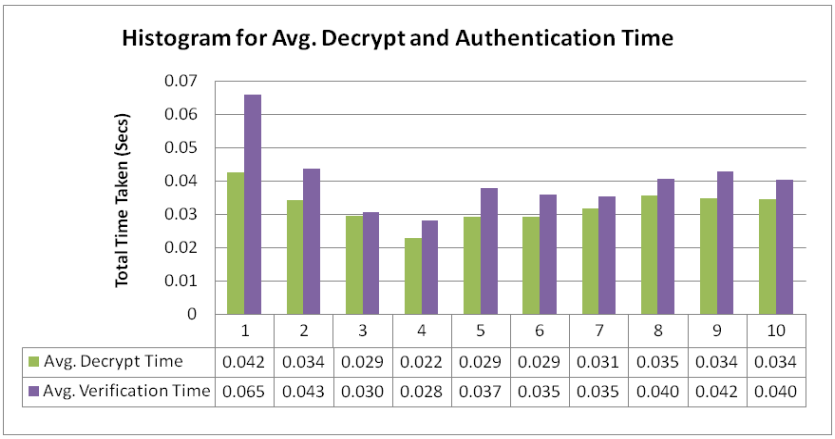

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| Execution Time | 0.25 | 0.38 | 0.48 | 0.51 | 0.65 | 0.75 | 0.81 | 1.14 | 1.35 | 1.65 |
| Avg. Reg. Exe. Time | 0.25 | 0.19 | 0.16 | 0.12 | 0.13 | 0.12 | 0.11 | 0.14 | 0.15 | 0.16 |

**Fig. 4.** Analysis of Registration Execution and Avg. Execution Time




| | 10 | 20 | 30 | 40 | 50 | 60 | 70 | 80 | 90 | 100 |
|---|---|---|---|---|---|---|---|---|---|---|
| Decrypt Time | 0.425 | 0.681 | 0.881 | 0.914 | 1.456 | 1.758 | 2.218 | 2.845 | 3.135 | 3.458 |
| Verification Time | 0.657 | 0.874 | 0.914 | 1.125 | 1.891 | 2.144 | 2.472 | 3.241 | 3.847 | 4.014 |

**Fig. 5.** Analysis of Decrypt and Authentication Execution Time



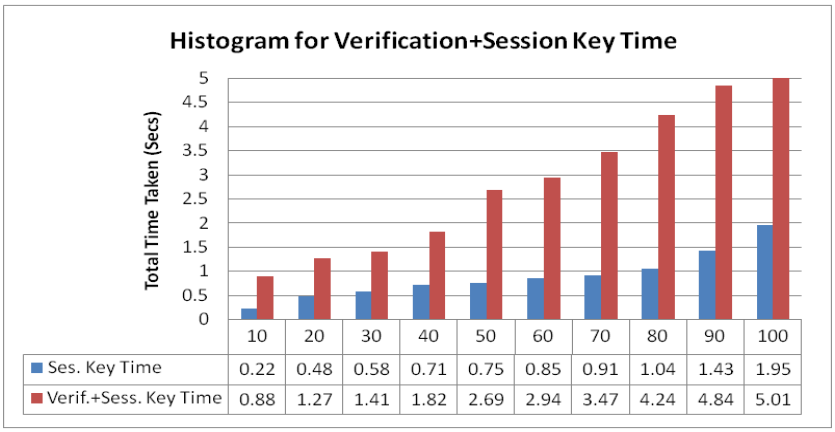

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| Avg. Decrypt Time | 0.042 | 0.034 | 0.029 | 0.022 | 0.029 | 0.029 | 0.031 | 0.035 | 0.034 | 0.034 |
| Avg. Verification Time | 0.065 | 0.043 | 0.030 | 0.028 | 0.037 | 0.035 | 0.035 | 0.040 | 0.042 | 0.040 |

**Fig. 6.** Analysis of Avg. Decrypt and Authentication Execution Time

**Session Key Generation Time.** We analysis here the total time taken for session key generation and the total time taken for complete authentication and session key generation. To do so we iterate the number user request from 10 to 100. The Fig. 7 shows the total time is taken for Session key generation and total time taken for complete verification, and Fig. 8 shows the avg. of this two comparison.

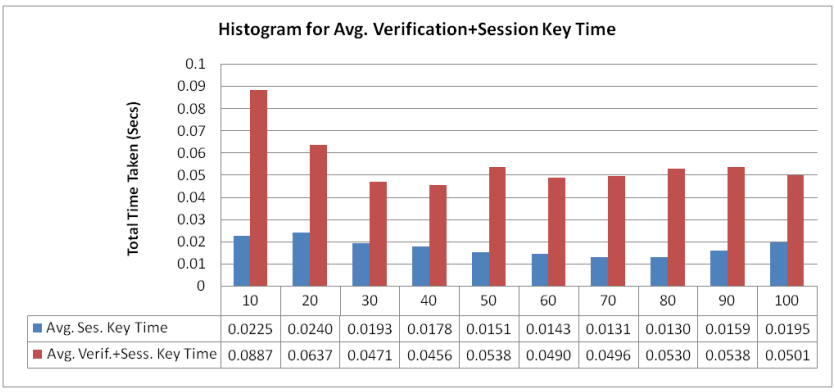
**Service Request Performance.** It measures the response time by the server for each request made per thread node. Every 1 second elapsed a node switch to new URL dynamically to evaluate the response time changes by the server. Fig. 9 shows the result of 10 users making 50 requests each.

All the obtained results in Fig. 4 to Fig. 9 confirms the efficiency of the proposal through reducing the execution time against the encryption and decryption process for the authentication and the session key generation. It suggests that it can support in reducing the computation cost and save the mobile device resources.



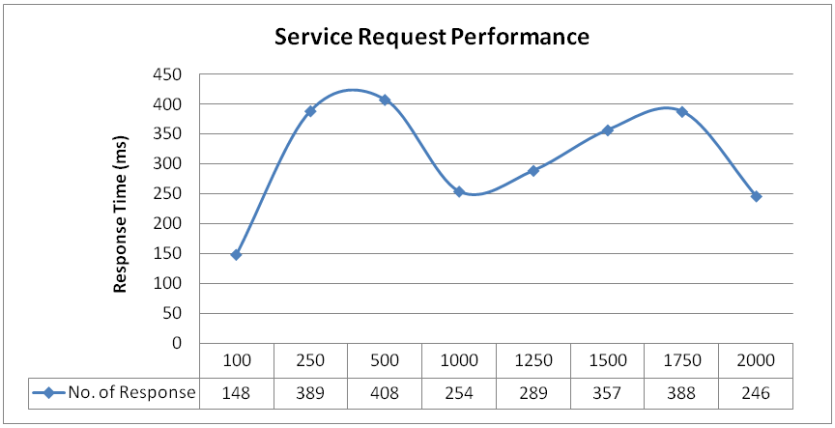

| | 10 | 20 | 30 | 40 | 50 | 60 | 70 | 80 | 90 | 100 |
|---|---|---|---|---|---|---|---|---|---|---|
| Ses. Key Time | 0.22 | 0.48 | 0.58 | 0.71 | 0.75 | 0.85 | 0.91 | 1.04 | 1.43 | 1.95 |
| Verif.+Sess. Key Time | 0.88 | 1.27 | 1.41 | 1.82 | 2.69 | 2.94 | 3.47 | 4.24 | 4.84 | 5.01 |

**Fig. 7.** Analysis of verification and Session key Execution Time




| | 10 | 20 | 30 | 40 | 50 | 60 | 70 | 80 | 90 | 100 |
|---|---|---|---|---|---|---|---|---|---|---|
| Avg. Ses. Key Time | 0.0225 | 0.0240 | 0.0193 | 0.0178 | 0.0151 | 0.0143 | 0.0131 | 0.0130 | 0.0159 | 0.0195 |
| Avg. Verif.+Sess. Key Time | 0.0887 | 0.0637 | 0.0471 | 0.0456 | 0.0538 | 0.0490 | 0.0496 | 0.0530 | 0.0538 | 0.0501 |

**Fig. 8.** Analysis of Avg. verification and Session key Execution Time




| | 100 | 250 | 500 | 1000 | 1250 | 1500 | 1750 | 2000 |
|---|---|---|---|---|---|---|---|---|
| No. of Response | 148 | 389 | 408 | 254 | 289 | 357 | 388 | 246 |

**Fig. 9.** Analysis of Service Request Performance

## 5 Conclusion

In this paper, we discuss the disadvantages of the existing user authentication scheme for the disturbed computer network. We propose a novel authentication mechanism to prevent unauthorized service access to mobile devices in distributed networks. This mechanism implements user authentication, user service session key generation, and service key exchange and validation methods. We implement a secure SSO mechanism to overcome these prospective drawbacks that may be particularly useful for mobile devices. When comparing our results analysis, the proposed method is better suited for mobile users using battery limiting devices because of their low computational cost and low communication cost. It can also apply to a distributed computer network where users are positioned with better performance in different services without adding a time synchronization mechanism.

## 6 References

[1] S. D. Yalew, P. Mendonça, G. McGuire, S. Haridi, M. Correia, "TruApp: A TrustZone-based Authenticity Detection Service for Mobile Apps", Proceedings of the 13th IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), 2017.

[2] R. Peeters, K. Grenman, "n-Auth: Mobile Authentication Done Right", In Proceedings of ACSAC 2017, USA, Pgs. 4–8, December-15, 2017. https://doi.org/10.1145/3134600.3134613

[3] D. Davidson, Y. Chen, F. George, L. Lu, S. Jha, "Secure Integration of Web Content and Applications on Commodity Mobile Operating Systems", In Proceedings of the ACM on Asia Conference on Computer and Communications Security Pages 652-665, 2017. https://doi.org/10.1145/3052973.3052998

[4] S. Ruoti, K. Seamons, End-to-End Passwords, In Proceedings of New Security Paradigm Workshop, Islamorada, Florida, USA, (NSPW'17), 14 pages, 2017.

[5] C.-H. Ling, C.-C. Lee, C.-C. Yang, and M.-S. Hwang1, "Secure and Efficient One-time Password Authentication Scheme for WSN", International Journal of Network Security, Vol.19, No.2, PP.177-181, Mar. 2017.

[6] J. Costa and A. Michalas, "Middle Man: An Efficient Two-Factor Authentication Framework", In 3rd IEEE International Conference On Computing, Communication, Control And Automation, 2017 https://doi.org/10.1109/ICCUBEA.2017.8463686

[7] N. Naik, P. Jenkins, D. Newell, "Choice of suitable Identity and Access Management standards for mobile computing and communication", IEEE 24th International Conference on Telecommunications (ICT), Pgs. 1 - 6, 2017 https://doi.org/10.1109/ICT.2017.7998280

[8] U. Shafique, A. Sher, R. Ullah, H. Khan, A. Zeb, et al., "Modern Authentication Techniques in Smart Phones: Security and Usability Perspective", International Journal of Advanced Computer Science and Applications(IJACSA), Vol. 8, No. 1, 2017. https://doi.org/10.14569/IJACSA.2017.080142

[9] G. Sharma, S. Bala, A. K. Verma, "An improved RSA-based certificateless signature scheme for wireless sensor networks", International Journal of Network Security, vol. 18, no. 1, pp. 82-89, 2016.

[10] C.-C. Chang and C.-Y. Lee, "A secure single sign-on mechanism for distributed computer networks", IEEE Transactions on Industrial Electronics, Vol. 59, Iss. 1, Pgs. 629-637, 2012.

[11] G. Wang, J. Yu, Q. Xie, "Security Analysis of a Single Sign-On Mechanism for Distributed Computer Networks", IEEE Transactions on Industrial Informatics, Vol. 9, Iss. 1, Pgs: 294 - 302, 2013.
[12] P. Mutchler, A. Doupe, J. Mitchell, C. Kruegel, G. Vigna, "A Large-Scale Study of Mobile Web App Security", In Proceedings of the Mobile Security Technologies Workshop (MoST). 2015.
[13] J. Yu, G. Wang, Y. Mu., □"Provably secure single sign-on scheme in distributed systems and networks. In. Proc. 11th IEEE International Conference On Trust, Security and Privacy in Computing and Communication (TrustCom□f12), pp 271-278, 2012.
[14] J. Zhang, Z. Wu, Y. Li, "An improved identity-based authenticated key agreement protocol using pairings", IEEE Proceedings of International Conference on Computer Science and Network Technology, Vol. 1, Pages: 45 - 49, 2011.
[15] J.-Z. Lu, J. Zhou, "On the Security of an Efficient Mobile Authentication Scheme for Wireless Networks", IEEE 6th International Conference on Wireless Communications Networking and Mobile Computing (WiCOM), Pg. 1 - 3, 2010. https://doi.org/10.1109/WICOM.2010.5601341
[16] C. W. Lin, C. S. Tsai and M. S. Hwang, □"A New Strong-Password Authentication Scheme Using One-Way Hash Functions", Journal of Computer and Systems Sciences International, vol. 45, no. 4, pp. 623-626, 2006. https://doi.org/10.1134/S1064230706040137
[17] E. Bresson, O. Chevassut, and D. Pointcheval, "Provably-Secure Authenticated Group Diffie-Hellman Key Exchange", ACM Transactions on Information and System Security, Vol. 10, No. 3., Pp. 255-264, 2007. https://doi.org/10.1145/1266977.1266979
[18] M. Pirker and D. Slamanig, "A framework for privacy-preserving mobile payment on security-enhanced arm trust zone platforms", In Proceedings of the 11th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, pp. 1155–1160, 2012.
[19] G. Yang, C. Tan, "Strongly secure certificateless key exchange without pairing", In 6th ACM Symposium on Information, Computer, and Communications Security, Page 71–79, 2011. https://doi.org/10.1145/1966913.1966924
[20] D. Pointcheval, M. Abdalla,D. Bernstein and T. Lange Eds, "Flexible Group Key Exchange with On-Demand Computation of Subgroup Keys", Springer-Verlag, LNCS 6055, pp.351-368, 2010.
[21] R. Cramer and V. Shoup, "Design and analysis of practical public-key encryption schemes secure against adaptively chosen ciphertext attack", SIAM Journal of Computing, vol. 33, no. 1, pp. 167–226, 2003. https://doi.org/10.1137/S0097539702403773
[22] C. Tang and D. O. Wu, "An efficient mobile authentication scheme for wireless networks", IEEE Transactions on Wireless Communications, vol. 7, no. 4, pp. 1408–1416, 2008. https://doi.org/10.1109/TWC.2008.061080
[23] K. Mangipudi, R. Katti, "A Secure Identification and Key agreement protocol with user Anonymity(SIKA)", Elsevier Computers & Security, Vol. 25, Issue 6, September 2006, Pages 420-425, 2006.
[24] L. Chen and C. Kudla, "Identity-based authenticated key agreement from pairings", IEEE Computer Security Foundations Workshop, pp. 219-233, 2003. https://doi.org/10.1109/CSFW.2003.1212715
[25] J. Jacob, M.John, □"Security Enhancement Of Single Sign-On Mechanism for Distributed Computer Networks", International Journal of Modern Engineering Research (IJMER), Vol. 3, Issue. 3, pp-1811-1814, 2013.

[26] J. Wang, G. Wang and W. Susilo,□"Anonymous Single Sign-on Schemes Transformed from Group Signatures", IEEE 5th International Conference on Intelligent Networking and Collaborative Systems, Pages: 560 - 567, 2013.
[27] JavaScript Runtime Process time, Link: "https://nodejs.org/api/process.html #process".
[28] Apache jMeter, Link : "http://jmeter.apache.org".
[29] D. Pointcheval, M. Abdalla, "Distributed Public-Key Cryptography from Weak Secrets", Springer-Verlag, LNCS 5443, pp.139-159, 2009.
[30] R. Álvarez, L. Tortosa, "Analysis And Design Of A Secure Key Exchange Scheme", Elsevier Information Sciences, Vol. 179, pp. 2014-2021, 2009. https://doi.org/10.1016/j.ins.2009.02.008

## 7 Authors

Dr. **V. L. Pavani** is an Assistant Professor, Madanapalle Institute of Technology and Science, Madanapalle, A.P. India, email: pavaniveluru1@gmail.com

Article submitted 04 January 2018. Resubmitted 23 September 2018. Final acceptance 22 November 2018. Final version published as submitted by the authors.