International Journal For Advanced Research In Science & Technology
A peer reviewed international journal
www.ijarst.in
ISSN: 2457-0362

Dynamic Reciprocal Authentication Protocol for Mobile Cloud Computing

Author 1: Mrs. D.Srilatha reddy
(Assistant Hod, Department of Computer Science and Engineering, Sphoorthy Engineering College, Hyderabad.
Email: dyapa.sreelatha@gmail.com
Author 2: M.Rahul, B.Tech
(Student, Department of Computer Science and Engineering, Sphoorthy Engineering College, Hyderabad.
Email:mrahul1509@gmail.com
Author 3: R.Vinay kumar reddy, B.Tech
(Student, Department of Computer Science and Engineering, Sphoorthy Engineering College, Hyderabad.
Email: vinaykumarreddyv96@gmail.com
Author 4: B.Manish yadav, B.Tech
(Student, Department of Computer Science and Engineering, Sphoorthy Engineering College, Hyderabad.
Email:manishbadulla8114@gmail.com

***Abstract*—**
A combination of mobile and cloud computing delivers many advantages such as mobility, resources, and accessibility through seamless data transmission via the Internet anywhere at any time. However, data transmission through vulnerable channels poses security threats such as man-in-the-middle, playback, impersonation, and asynchronization attacks. To address these threats, we define an explicit security model that can precisely measure the practical capabilities of an adversary. A systematic methodology consisting of 16 evaluation criteria is used for comparative evaluation, thereby leading other approaches to be evaluated through a common scale. Finally, we propose a dynamic reciprocal authentication protocol to secure data transmission in mobile cloud computing. In particular, our proposed protocol develops a secure reciprocal authentication method, which is free of Diffie– Hellman limitations, and has immunity against basic or sophisticated known attacks. The protocol utilizes multifactor authentication of usernames, passwords, and a onetime password. The one-time password is automatically generated and regularly updated for every connection. The proposed protocol is implemented and tested using Java to demonstrate its efficiency in authenticating communications and securing data transmitted in the mobile cloud computing environment. Results of the evaluation process indicate that compared with the existing works, the proposed protocol possesses obvious capabilities in security and in communication and computation costs.

***Index Terms*— Mobile cloud computing, Authentication, Diffie– Hellman, One-time password.**

I. INTRODUCTION

Mobile cloud computing (MCC) is a combination of mobile and cloud computing. In general, MCC incorporates mobile computing, wireless networking, and cloud computing to provide cloud-based

International Journal For Advanced Research In Science & Technology
A peer reviewed international journal www.ijarst.in
ISSN: 2457-0362

services to mobile users. The advantages of MCC include mobility, real-time data

Abdulghani Ali Ahmed is with the Cyber Technology Institute, School of Computer Science and Informatics, De Montfort University, The Gateway, Leicester, LE1 9BH, United Kingdom. Ahmed is also with Safecyber Systems Corporation, Malaysia (aa.ahmed@ieee.org).

availability, ease of access, and convenience as users can access and manage their data and applications through the Ethernet or Internet anywhere and anytime regardless of heterogeneous environments and platforms [1]. In addition, MCC enables data storage and processing outside the mobile device [2]. Successful adoption of MCC necessitates robust and effective authentication solutions through which users can utilize cloudbased services from any mobile device with low computing cost on the native resources. Although MCC is beneficial, lack of strong security features is a critical factor that may hinder the utilization of this technology.

Accessing and utilizing remote cloud-based resources are accompanied with concerns in security and privacy, including authentication and authorization of mobile users. In general, the mobile devices are connected to the cloud-based resources through the insecure wireless channel. As mentioned in [3], the main security challenge in MCC is authenticating the identity of mobile users so that forgery attacks can be detected and prevented. In forgery attacks, hackers masquerade as real users, log in to their accounts, and perform unauthorized actions to steal sensitive data. The sensitive data may include users' credentials, identity, location, job, and biometrics stored on the mobile device.

To prevent identity forgery attacks in MCC, connections between mobile client (M_{Client}) and cloud server (C_{Server}) need to be authenticated. Any connection between M_{Client} and C_{Server} can be authenticated using one-way or mutual authentication. Although helpful, one-way authentication does not provide an absolute secure connection as authentication is performed on one side only, that is, M_{Client} authenticates C_{Server} or C_{Server} authenticates $M_{Client.}$ By contrast, mutual authentication is efficient because ideally, both parties communicating must prove their identity to each other. Lack of mutual authentication in MCC allows hackers to intercept the communication channel and manipulates messages that are transmitted between the C_{Server} and M_{Client}. Besides, mobile users are also vulnerable to impersonation because their sensitive data can be easily obtained through phishing, spyware, and social engineering using their mobile devices.

Kwan Wendy and Mohammed Noman Kabir are with the Faculty of Computing, Universiti Malaysia Pahang, 26600 Pekan, Pahang, Malaysia.

Ali Safaa Sadiq is with the Wolverhampton Cyber Research Institute, School of Mathematics and Computer Science, University of Wolverhampton, Wolverhampton, Wulfruna Street, WV1 1LY, UK.

Although many authentication schemes have been proposed in recent years [4][5][6][7][8][9][10][11][12][13][14], most of them lack mutual authentication between M_{Client} and $C_{Servers}$ [15][16]. Moreover, the existing schemes are vulnerable to known attacks such as man-in-the-middle (MITM), playback, impersonation, and asynchronization [17][18][19]. These attacks represent serious threats to the existing authentication protocols in which attackers can do more than observe, modify, and/or capture user credentials while transmitting between M_{Client} and C_{Server}. The attacker can also reuse the captured credentials and retransmit it at a later time for nefarious purposes such as circumventing authentication and creating duplicate connection [20].

In this study, to analyze the vulnerabilities of the existing schemes, we define a security model that can precisely capture the capabilities of the adversary in exploiting the vulnerabilities of these studies. The security model covers a set of 10 known attacks that

International Journal For Advanced Research In Science & Technology
A peer reviewed international journal www.ijarst.in
ISSN: 2457-0362

create potential threats to the existing authentication schemes. We then use a set of 16 evaluation criteria to rate the performance of the existing schemes in terms of their capabilities to resist the defined list of threats and in terms of the computation overhead and communication cost. As the main contribution to this research, a **d**ynamic and **r**eciprocal authentication protocol is proposed to secure the communication between $M_{Clients}$ and $C_{Servers}$ in **MCC** environment (DRmcc). DRmcc is reciprocal because it develops a secure mutual authentication method, free of Diffie–Hellman limitations, and immune to known attacks. It is dynamic because it uses a one-time password (OTP), which is automatically generated and regularly updated.

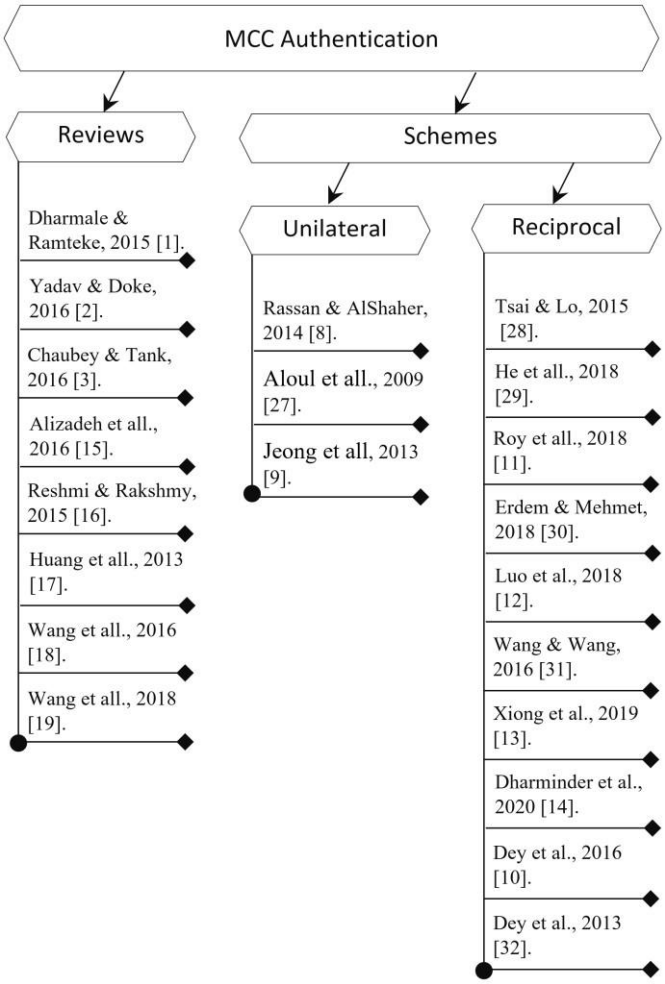
The DRmcc manages the reciprocity between $M_{Clients}$ and C_{Server} by applying a special set of rules in two phases: registration and connection. In the registration phase, the M_{Client} is registered to the cloud service provider using multi-factor passwords consisting of international mobile equipment identity (IMEI) number, username, and password. Upon obtaining the multi-factor passwords, the OTP is generated simultaneously at both M_{Client} and C_{Server} by concatenating the multi-factor passwords. In the connection phase, DRmcc starts working when the mobile requests establish a connection with the C_{Server}. Once the connection request is issued by the mobile device and received by the server, both the mobile device and server start to separately and simultaneously compute the Diffie–Hellman parameters to automatically update and encrypt (at the mobile client) or decrypt (at the cloud server) the OTP. Thus, the connection is established only when the OTP is matched.

This paper is structured as follows. Section 2 describes related works. Section 3 presents the proposed DRmcc protocol. Section 4 explains the communication model. Section 5 provides the threat model and evaluation criteria. In section 6, the experimental results are presented. In section 7, the performance of the proposed DRmcc protocol is evaluated. Finally, Section 8 concludes the paper and provides directions for future work.

II. RELATED WORKS

This section studies the state-of-the-art of research relevant to DRmcc. Numerous reviews have been conducted to analyze the advantages and disadvantages of the current studies in MCC authentication [17][18][19]. An observational study was conducted in [17] to analyze and examine the efficiency of two smart card-based password authentication schemes [21][22]. This study emphasizes that authentication schemes, which rely on smart card, are vulnerable to dictionary attacks. It concludes that the computation of session keys is possible where the attacker performs a password dictionary attack to obtain the user's password; by eavesdropping on the communication channel, the attacker can obtain the user ID and pre-computed hash keys stored in the smart card. Using these parameters, the attacker can calculate the session key and use it to decrypt transmitted messages. Moreover, the password that consists of eight characters and is selected from the human memorable domain is more vulnerable to dictionary attacks.

Wang *et al*. [18] reviewed three mobile device authentication schemes [23][24][25] and presented the challenges that the researchers face in designing an authentication scheme for mobile device, preserving the user's anonymity and privacy. One of the challenges is that the mobile device authentication scheme is vulnerable to known sessionspecific attacks where temporary information stored in the mobile device is leaked due to improper memory clean-up or obtained through side-channel attacks. The usage of long-term private keys and usernames/passwords within the human memorable domain also makes the mobile device authentication scheme vulnerable to key-compromise impersonation attacks. Another security threat is collusion attack, where the attacker colludes with a legitimate foreign server to disclose the credentials of the mobile user. A systematic framework to evaluate the two-factor authentication scheme is conducted in [19]. The conducted framework concludes by discontinuing



the break–fix–break–fix cycle in the research domain of two-factor authentications.

In addition to the review studies, the state-of-the-art section reviews the most recent approaches and schemes proposed to enhance the MCC authentication. We classify these proposed approaches and schemes into two categories, namely, unilateral and reciprocal authentication [26], as illustrated in Fig. 1. The following subsections describe these categories in detail.

A. Unilateral Authentication

This category is a one-way authentication performed at one end of the connection (either sender or receiver). Studies in this category focus on checking the authenticity at M_{Client} or C_{Server}. A biometric authentication mechanism that uses fingerprint recognition systems to secure mobile cloud computing [8] falls under this category. This mechanism employs existing cameras in mobile phones to capture the fingerprint image of a cloud user. Then, the captured image is sent to a core-point detection phase where feature extraction of the fingerprint image is conducted. Finally, the user is verified and authenticated to the C_{Server} if the extracted fingerprint image matches the one stored in the database. In addition to its one-way authentication, this mechanism has a high cost because it requires a high-quality camera to capture an accurate fingerprint image.

Fig. 1. Classification of Authentication Schemes in Mobile Cloud Computing

Another one-way authentication study was conducted by [27]. This study proposed a multi-factor authentication method for generating an OTP and an additional SMS-based authentication system. The OTP generated in this study uses a set of factors such as username, password, IMEI, and international mobile subscriber identity (IMSI), which are concatenated and hashed using SHA-256. The SMS-based system serves as a back-up mechanism and as a means of synchronization. The proposed method reduces the organizational cost of purchasing and maintaining hardware tokens by using software tokens for verification. However, the utilized one-time password is not encrypted and service charges are incurred when the SMS-based authentication is used.

Jeong *et al*. [9] proposed an authentication system for smart devices using multiple factors in a mobile cloud service. The system uses ID/password, IMEI, IMSI, voice recognition, and face recognition as authentication parameters. The system uses the management server to perform load balancing. The load is sent to a clustered host of virtual machines to authenticate the information given by the mobile user. The result of the authentication process is sent to a management server, which returns the final authentication result with the user's authentication values to the smart device. This system enhances authentication performance because the C_{Server} processes the factors in bulk, but no reciprocal authentication occurs between mobile users and the C_{Server}. The system also lacks usability and privacy because it requires multiple types of sensitive data.

Reciprocal Authentication

This category is a mutual authentication performed by both

International Journal For Advanced Research In Science & Technology
A peer reviewed international journal www.ijarst.in
ISSN: 2457-0362

M_{Client} and C_{Server} at the two ends of the connection. Under this category, a private authentication scheme conducted in [28] uses a smart card generator (SCG). The scheme applies dynamic nonce generation and bilinear pairing cryptosystem techniques. This scheme reduces the complexity of discrete logarithm problems. Mobile users or service providers register to the SCG by providing their information while the SCG computes and securely sends the respective private keys to MClient and CServer. When MClient and CServer want to communicate, a card provided by the trusted SCG is used to authenticate both of them. Although this scheme is conducted to support mutual authentication, an attacker can still impersonate C_{Server} to M_{Client}. Also, the attacker can extract M_{Client}'s real identity while executing the C_{Server} impersonation attack [29]. Another limitation of this scheme is the risk of losing the card, which is essential for both M_{Client} and C_{Server} to authenticate each other.

The security limitations in [28] are addressed by a recent scheme [29], which constructs privacy-aware authentication for MCC services by using an identity-based signature scheme. As this scheme is constructed based on the SCG scheme [28], it still inherits security limitations such as the inability to resist impersonation attacks and stolen smart card attacks.

A combined approach of fine-grained data access control over distributed cloud servers using mobile user authentication mechanisms is proposed in [11]. In particular, this scheme is proposed to control mobile users' privileges relevant to accessing the data stored in the cloud-based multi-server. This approach ensures that both parties of cloud server and mobile users are verified before generating a permission key and shared session key required to access the data stored in the cloud server. However, this scheme is vulnerable to asynchronous attacks where an attacker can delay the transmitted message intentionally beyond the acceptable time, causing both parties to fail the authentication and authorization process [13].

An approach to using the OTP as a service has been proposed in [30]. This method describes an architecture between service provider, cloud user, and one-time password provider. The proposed architecture is not intended to solve a traditional username or password, but adding a second factor to traditional authentication offers a stronger and more efficient authentication process. In this approach, the user is expected to run the private key exchange phase for every service used in the cloud [15]. This approach is still lacking in usability because users are expected to remember the characters of the OTP and type them within a given period for authentication purposes.

A three-factor-based authentication scheme for real-time data access in wireless sensor networks (WSNs) is proposed in [12] to provide higher security and operational efficiency when compared with the two-factor-based authentication schemes. The proposed scheme uses smart card, biometric information and user's password and username factors to provide an authenticated real-time access to data in WSNs. This scheme is resistant to password/biometric key guessing attacks, replay attacks, clone card attacks, node capture attacks, and protects user/sensory anonymity besides providing mutual authentication. However, this scheme is vulnerable to asynchronous attacks as it uses time stamps to validate the transmitted messages between all parties. This scheme may allow attackers to delay the message transmission, thereby causing failure to authenticate between the user, gateway, and sensor nodes [13]. Moreover, the three-factor scheme involves a high number of operations that may cause an extra computation overhead. The extra overhead along with sensor nodes that has limited memory may lead to a reduction in the efficiency of the proposed scheme [15].

A smart-card-based password authentication scheme is proposed in [31] as an alternative solution for the two-factor authentication scheme. Although helpful, this scheme depends on smart cards and therefore, it may inherit the limitations of the smart card-based schemes as described in [17].

International Journal For Advanced Research In Science & Technology
A peer reviewed international journal www.ijarst.in
ISSN: 2457-0362

Reference [13] introduced a lightweight anonymous authentication scheme with forward secrecy (LAASF) that is resistant to security threats such as asynchronization attack and smart card loss attack (SCLA). LAASF is formed to be resistant to security attacks such as smart card loss and replay attack. However, the authentication scheme involved a system of three parties where an external user has a smart card, a sensor node, and a gateway node. Thus, LAASF requires authentication among these three parties. Moreover, LAASF uses the same secret key of gateway node (GWN) and long-term secret key between the user and GWN in its authentication processes.

An RSA-based authentication scheme [14] has been proposed for use in healthcare service, where it can resist password guessing and ensure key agreement during the exchange of two messages. Two-factor authentication is used in this scheme, which requires a user's ID, a password, and a smart card. The scheme uses timestamp and hash keys XOR with a random value to send messages to the server for verification and vice versa. The scheme is said to resist various attacks such as insider attack, password guessing, stolen smartcard attack, and impersonation attack. It can also preserve user anonymity, unlinkability, and secure the session key. However, the scheme uses timestamps to verify valid messages in the authentication process; therefore, it inherits the limitation described in [11], where the scheme is vulnerable to an asynchronous attack that causes delayed messages and failure of authentication between client and server.

Most of the work related to the DRmcc protocol is the message digest-based authentication (MDA) scheme [10][32]. The MDA scheme consists of two phases: one where the C_{Server} authenticates the M_{Client} and another in which the M_{Client} authenticates the C_{Server}. Although this scheme provides mutual authentication, the authentication operations involve many processes such as the generation of random and authentication keys, hashing of message digests, and encryption and decryption of the message digest, which is performed in both parties. Moreover, a large number of messages are transmitted between the M_{Client} and C_{Server}, which makes the MDA scheme less efficient [15]. Aside from that, as the MDA scheme utilizes the standard Diffie–Hellman algorithm, it is vulnerable to MITM attacks, which may be launched to sniff
encryption/decryption keys during the process of private– public key distribution.

The single/multi-factor authentication schemes reviewed in this paper have merits and limitations, which depend on the capability or incapability to resist the various attacks that the adversaries may use to gain an unauthenticated connection. These schemes are resistant to most but not to all of the attacks. The schemes proposed in [9][11][12][14] are not resistant to asynchronous attacks. Schemes in [10][31][32] are vulnerable to playback attacks [20][29]. Given that these methods encrypt credentials of M_{Client} before transmitting them to the C_{Server}, these methods may be safe against capturing and modifying credentials but are not immune to replay attacks. In replay attacks, attackers are able to capture the credentials and reuse them to establish a new connection even if the credentials are encrypted once transmitted. Moreover, schemes that are proposed in [9][10][11][32] are vulnerable to shoulder surfing attacks. In addition to the limitations discussed with every method, it should be stated here that all schemes [9][10][11][12][13][14][32] have scalability shortcoming, as they require high communication cost as well as high computation overhead.

The DRmcc protocol mitigates the limitations of unilateral authentication methods by conducting mutual authentication at M_{Client} and C_{Server}. This protocol also alleviates the limitations of the reciprocal authentication methods by proposing a lightweight method to reduce the number of processes and provide scalable communication between M_{Client} and C_{Server}. Although the DRmcc protocol partially utilizes the Diffie– Hellman structure, it has a significant contribution to prevent the MITM attack, which may be inherited from Diffie– Hellman. Moreover, the DRmcc protocol is secure against impersonation, replay,

International Journal For Advanced Research In Science & Technology
A peer reviewed international journal
www.ijarst.in
ISSN: 2457-0362

and asynchronization attacks by using a different OTP for every connection. Furthermore, the DRmcc is resistant to shoulder surfing attacks as the OTP is dynamically and automatically generated by the M_{Client} and C_{Server} without the need to be keyed in by the users.

III. DRMCC AUTHENTICATION PROTOCOL

This section describes the proposed protocol DRmcc and its mutual multi-factor authentication scheme. The DRmcc consists of registration and connection phases, as illustrated in Fig. 2. These two phases are described in the following subsections.

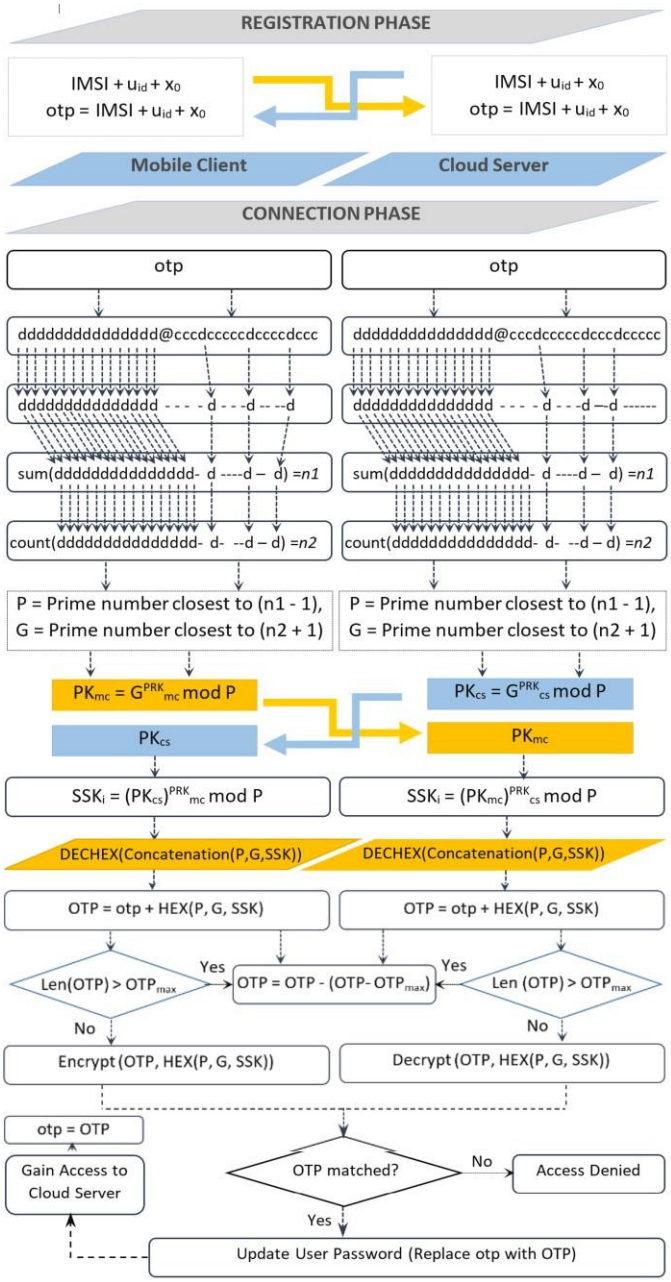


Fig. 2. DRmcc authentication protocol.

A. Registration Phase

Two ends are involved in the authentication process, namely, mobile (M_{Client}) end and cloud provider (C_{Server}) end. In the registration phase, the M_{Client} requests to register as client to the C_{Server}. Thus, it is required to set up an account on the C_{Server} by registering its username, password, and IMEI metrics. In this protocol, the metrics of registration can be exchanged between the M_{Client} and C_{Server}

International Journal For Advanced Research In Science & Technology
A peer reviewed international journal www.ijarst.in
ISSN: 2457-0362

using out-of-band authentication method such as SMS to strengthen immunity against MITM attacks. Therefore, the probability of sniffing these metrics and using them to spoof the identity of one of the connection ends is not considerable. In addition to the authenticity of the utilized outof-band method, the protocol does not present a considerable extra communication overhead because it is made for one time only at the beginning of the registration phase. Moreover, to arrange for a reciprocal authentication during the connection phase, a one-time password (OTP) is generated as a concatenation of the username, password, and IMEI metrics. These metrics along with the OTP are saved in a small database in MClient and CServer.

B. Connection Phase

Connection phase starts when M_{Client} requests to connect and gain access to C_{Server}. To get connected, both C_{Server} and M_{Client} simultaneously implement a set of rules in order to authenticate the communication between each other as shown in Fig 2. The implemented set of rules represents the DRmcc authentication protocol proposed in this paper. The DRmcc protocol is executed separately at the C_{Server} and M_{Client}. The authentication process depends on using an OTP-generated instantly for every connection. In the DRmcc protocol, each C_{Server} and M_{Client} uses the former OTP saved from the previous connection to generate an instant OTP to be used in a new connection. For the first time of connection, both C_{Server} and M_{Client} use the OTP saved during the registration phase. In general, maintaining the reciprocal authenticity of the two ends of connections in the connection phase is achieved through two main steps: generating an instant OTP and encrypting or decrypting the instant OTP.

The instant OTP is automatically generated by using the former OTP and Diffie–Hellman-shared parameters, particularly a prime number (P), a generator number (G), and the secret session key (K). In DRmcc, the values of P and G are generated from the former OTP content. The generation process is made by extracting the numerical digits from the content of the former OTP. The P value is generated by computing the sum of all the numerical digits included among the contents of the former OTP. The G value is generated by counting the total number of numerical digits included in the content in the former OTP. According to Diffie–Hellman, the values of P and G should be prime numbers. Thus, if the computed value of P is not a prime number, the closest prime number greater than the current value of P is calculated and used as a prime value for P. Likewise, if the value of G is not a prime, the closest prime number smaller than the current value of G is calculated and used as a prime value for G.

In encrypting or decrypting the instant OTP, the Diffie– Hellman algorithm is used to encrypt (at M_{Client} end) or decrypt (at C_{Server} end) the instant OTP generated in the previous step. The parameters required to generate a shared secret session key (SSK) at M_{Client} and C_{Server} using Diffie–Hellman are P, G, public key of mobile client PK_{mc}, and public key of C_{Server} PK_{cs}. In this regard, the values of P and G need not be exchanged between M_{Client} and C_{Server} because these values have been automatically and separately computed in advance at the mobile and cloud sides. The public keys PK_{mc} and PK_{cs} are generated based on the Diffie–Hellman algorithm. The mobile then sends its public key PK_{mc} to C_{Server}, and C_{Server} sends its public key PK_{cs} to M_{Client}. Exchange PK_{mc} and PK_{cs} between M_{Client} and C_{Server} is the unique exchange process in the DRmcc protocol. The key advantage of DRmcc is that if it happens that PK_{mc} and PK_{cs} are sniffed using MITM attack, it does not affect the authenticity of DRmcc as the SSK cannot be computed without knowing the other parameters of Diffie–Hellman such as P, G, private random key of mobile client PRK_{mc}, and private random key of C_{Server} PRK_{cs}. Moreover, in the DRmcc protocol, values of the Diffie–Hellman parameters are not constant; they are updated and changed for every connection, which makes the SSK immune to hacking attempts.

International Journal For Advanced Research In Science & Technology
A peer reviewed international journal www.ijarst.in
ISSN: 2457-0362

Once the public keys PK_{mc} and PK_{cs} are computed using Diffie–Hellman method, the M_{Client} and C_{Server} exchange the public keys with each other to compute the Diffie–Hellman shared SSK. Upon receiving the PK_{mc} and PK_{cs} from the client and server, calculation to obtain the shared SSK is performed in the client and server simultaneously. Once the SSK number is generated, its value is concatenated with values of P and G parameters to generate the new OTP.

Given that the instant OTP is generated by concatenating the values of the former OTP, P, G, and SSK, two consequences should be considered to maintain the usability of the DRmcc protocol. The first consequence is relevant to the OTP characters where after a few connections, the entire OTP content becomes digits only. Thus, the value of G is fixed for the new connections. DRmcc protocols mitigate this consequence by converting the values of P, G, and SSK into hexadecimal values before concatenating them to the former OTP. In this way, the instant OTP is guaranteed to include the mixed content of digits and characters and thus, the value of G remains variable continuously. The second consequence is the length of the OTP, where after a number of connections, the length of the instant OTP may become excessive. Thus, the length of the instant OTP is checked with every connection to verify if it exceeds a predefined maximum length of OTP (OTP_{max}). The OTP_{max} is set as 16, 32, or 64 characters based on the demands of the administrator. If the length of the instant OTP exceeds the predefined maximum length, then the length of the instant OTP is reduced by deducting the outcomes of the predefined maximum length (OTP_{max}) from the current length of the instant OTP.

The shared SSK is then used as a symmetric key for encrypting (at the M_{Client} side) and decrypting (at the C_{Server} side) the instant OTP. The final authentication process starts when the M_{Client} encrypts the instant OTP using the computed SSK and sends it to the C_{Server}. In turn, the C_{Server} verifies the authenticity of the M_{Client} in two processes. The first process is by decrypting the received OTP using the shared SSK computed at the C_{Server}. The second process is by comparing the decrypted OTP with the instant OTP, which is generated at the server. If the OTPs generated at the M_{Client} matches the one generated at the C_{Server}, then the server and mobile establish the communication session and start the data transmission. Once the connection is successfully established, both M_{Client} and C_{Server} update the old OTP stored in their database with the recent OTP used in the active connection. The stored OTP is not updated in case the connection between M_{Client} and C_{Server} is not successfully established for any reason.

II.COMMUNICATION MODEL

The communication model of DRmcc is described by Algorithm 1. In the algorithm, first a M_{Client} is registered to the C_{Server} by passing the mobile identity number $\boldsymbol{M_{id}}$, a user identity number $\boldsymbol{u_{id}}$, and an initial password x_0. For each connection request i, exchange keys A (on the client side), B (on the server side), and the encrypted message $\boldsymbol{M_{E,i}}$ is formulated. The terms A and B are calculated on the client side and the server side by

$$A = ([m]^p)^{a_i} \bmod \left\lfloor \sum^{m} S_k \right\rfloor$$
$$B = ([m]^p)^{b_i} \bmod \left\lfloor \sum_{k=1} S_k \right\rfloor_p \quad (1)$$

where S is the set of all numeric characters in F_j i.e., $\boldsymbol{S} = \{\cup_{k \in K} \boldsymbol{F_{jk}} \mid \boldsymbol{F_{jk}} \in \mathbb{N} \textbf{ where } \boldsymbol{K} = \{\boldsymbol{1}, \boldsymbol{2}, \ldots, \boldsymbol{n}\}\}$ and n is the length of F_j, m is the length of S_k, a_i is the secret key generated by the client at connection request i, $\lfloor \boldsymbol{z} \rfloor_{\boldsymbol{p}}$ is the largest prime number less than z, and $\lceil \boldsymbol{z} \rceil^{\boldsymbol{p}}$ is the smallest prime number larger than z. Then, considering encryption and decryption processes, the value of the key $\boldsymbol{K_i}$ can be deduced using

International Journal For Advanced Research In Science & Technology
A peer reviewed international journal www.ijarst.in
IJARST
ISSN: 2457-0362

$$K_i = \left[\left([m]^p \bmod \left| \sum_{k=1}^{m} S_k \right|_p \right)^{a_i b_i} \right] \bmod \left| \sum_{k=1}^{m} S_k \right|_p \quad (2) \quad (2)$$

Thus, we check the encryption and decryption of a certain F_j result in the same key value. Finally, the encrypted message as a one-time password is formulated as follows:

$$M_{E,i} = E\,(f\,(M_{id}, u_{id}, x_j), K_i),$$

$$= E\,(f\,(M_{id}, u_{id}, h\,(x_{j-1}, K_{i-1})), K_i), \quad (3)$$

where $E(.)$ is an encryption function, and $h(.)$ is a passwordupdate function. The encrypted message depends on all the previous passwords and generated keys, which add the level of complexity to detect the password. The decryption process can be presented by $F_D = D(M_{E,i}, K_i)$, where $D(.)$ is the decryption function.

IV. THREAT MODEL AND EVALUATION CRITERIA

This section describes a realistic adversary model [33][34][35], which explicitly defines the capabilities of the attacker that threatens the proposed DRmcc protocol. A set of 10 criteria used is also presented to evaluate the performance of the DRmcc protocol compared with the existing works. Both the adversary model and evaluation criteria are further described in the following subsections.

ALGORITHM 1. DRmcc communication model.

Mobile client	Cloud server
1. Send $F_0 = f(M_{id}, u_{id}, x_0)$ to cloud for an initial registration.	1. Receive $F_0 = f(M_{id}, u_{id}, x_0)$ and register M_{id}, u_{id}, x_0.
2. Set $j := 0$.	2. Set $j := 0$
3. **for** each connection request $i = 0, 1, 2,\ldots$	3. **for** each connection request $i = 0, 1, 2,\ldots$
4. Find $S \subseteq F_j$ where $S = \{\cup_{k\in K} F_{jk} \mid F_{jk} \in \mathbb{N}$ **where** $K = \{1, 2, \ldots, n\}\}$	4. Find $S \subseteq F_j$ where $S = \{\cup_{k\in K} F_{jk} \mid F_{jk} \in \mathbb{N}$ **where** $K = \{1, 2, \ldots, n\}\}$
5. Compute $P = \lvert\sum_{k=1}^{m} S_k\rvert_p$ and $Q = \lvert m\rvert^p$	5. Compute $P = \lvert\sum_{k=1}^{m} S_k\rvert_p$ and $Q = \lvert m\rvert^p$
6. Use a dynamic secret key a_i	6. Use a dynamic secret key b_i
7. $A = Q^{a_i} \bmod P$	7. $B = Q^{b_i} \bmod P$
Send A	Send B
8. Receive B from cloud	8. Receive A from client
9. Compute $K_i = B^{a_i} \bmod P$	9. Compute $K_i = A^{b_i} \bmod P$
10. Compute new password: $Y = h(x_j, K_i)$;	10. Compute new password: $Y = h(x_j, K_i)$;
11. Compute OPT $F_j = f(M_{id}, u_{id}, Y)$	11. Compute OPT $F_j = f(M_{id}, u_{id}, Y)$
12. Encrypt: $M_{E,i} = E(F_j, K_i)$	12. Receive $M_{E,i}$ as one time password and
Send $M_{E,i}$ as one time password	Decrypt: $F_D = D(M_{E,i}, K_i)$
13. **if** communication is established, then	13. **if** $F_D = F_j$, then
14. Update the terms: $x_{j+1} = Y$; $F_{j+1} = F_j$	Communication is established.
15. Set j := j +1;	14. Update the terms: $x_{j+1} = Y$; $F_{j+1} = F_j$
16. **end** (if)	15. Set j := j +1
17. **end** (for)	16. **end** (if)
	17. **end** (for)

A. Adversary Model

Defining an adversary model is necessary to assess the security of the proposed DRmcc protocol. To this end, the following describes the main capabilities of adversary $\bar{A}$ in DRmcc:

1) Adversary $\bar{A}$ is able to sniff the values of the parameters P and G shared between the M_{Client} and C_{Server} as well as in full control of the exchanged values of PK_{mc}, and PK_{cs}.
2) Adversary $\bar{A}$ can capture the data exchanged between M_{Client} and C_{Server} and reuses the captured credentials at a later time to duplicate the connection and obtain access to the system.
3) Adversary $\bar{A}$ may obtain and analyze the OTP to know the identity of the sender and receiver or to link messages.
4) Adversary $\bar{A}$ is able to practice offline guessing for all the parameters of P, G, PKmc, and PKcs at M_{Client} or C_{Server}. Thus, the adversary is able to guess the OTP offline.
5) Adversary $\bar{A}$ is able to retrieve the previously generated session key(s).
6) Adversary $\bar{A}$ may obtain and analyze the previously utilized OTP to generate a new OTP to be used in establishing an illegal connection with M_{Client}.
7) Adversary $\bar{A}$ is able to obtain and analyze the previously utilized OTP to generate a new OTP to be used in establishing an illegal connection with C_{Server}.
8) Adversary $\bar{A}$ is capable of releasing the OTP by exploiting the delay between the time of creating that OTP and the time of using it. This way, adversary $\bar{A}$ can use the created OTP before it is used by M_{Client} or C_{Server}.
9) Adversary $\bar{A}$ may watch over the victim's shoulder to nab the OTP during the time it is being keyed into an electronic device. Thus, the adversary is able to steal the identity of the victim, which can be either M_{Client} or C_{Server}.
10) Adversary $\bar{A}$ is able to recover the OTP from the SIM card of M_{Client} by practicing offline, online, or hybrid guessing.

International Journal For Advanced Research In Science & Technology
A peer reviewed international journal www.ijarst.in
ISSN: 2457-0362

B. Evaluation Criteria

We create our evaluation criteria set by considering the consistency with the evaluation criteria applied in the previous studies [19][31][36]. Our evaluation list covers 16 criteria that are essential to evaluate the performance of DRmcc in terms of security, usability, deployability, computation overhead, and communication cost that the DRmcc protocol satisfies:

1) Resistance to MITM attacks: The parameters required to generate a shared SSK at M_{Client} and C_{Server} in DRmcc are P, G, PK_{mc}, and PK_{cs}. In this regard, the values of P and G need not be exchanged between the M_{Client} and C_{Server} as these values have been automatically and separately computed in advance at the mobile and cloud sides. The public keys PK_{mc} and PK_{cs} are generated based on the Diffie–Hellman algorithm and exchanged between M_{Client} and C_{Server}. However, even if PK_{mc} and PK_{cs} are sniffed using MITM attack, it does not affect the authenticity of DRmcc because the SSK cannot be computed without knowing the other parameters of Diffie–Hellman such as P, G, PRK_{mc}, and PRK_{cs}. This way, the DRmcc protocol is not vulnerable to MITM attacks and can resist insider attacks.
2) Resistance to playback attack: The attacker reuses the captured credentials and retransmits them at a later time to duplicate the connection and gain access to the system. The DRmcc protocol utilizes a different OTP generated instantly for every connection. Each C_{Server} and M_{Client} uses the former OTP saved from the previous connection to generate a new and instant OTP for a new connection. Therefore, DRmcc is secure against playback attack where every generated password cannot be used for more than one connection.
3) Resistance to server impersonation attack: To impersonate C_{Server}, an adversary needs to decrypt the OTP, which the M_{Client} uses to request the new connection. To decrypt the OTP sent by the M_{Client}, the adversary requires obtaining the values of P, G, and SSK. The values of P, G, and SSK have been automatically and separately computed at the mobile and cloud sides without exchanging them between the M_{Client} and C_{Server}. In this manner, the adversary is unable to sniff the values of P, G, and SSK, and is therefore unable to practice the C_{Server} impersonation attack.
4) Resistance to client impersonation attack: To impersonate M_{Client}, an adversary needs to encrypt the OTP, which the C_{Server} uses to accept the new connection. To encrypt the OTP to be sent to C_{Server}, the adversary requires obtaining the values of P, G, and SSK. The values of P, G, and SSK have been automatically and separately computed at the mobile and cloud sides without exchanging them between M_{Client} and C_{Server}. In this way, the adversary is unable to sniff the values of P, G, and SSK, and therefore unable to practice the M_{Client} impersonation attack.
5) Anonymity and unlinkability: The authentication process in the DRmcc protocol completely depends on the OTP. The utilized OTP is generated separately and simultaneously at the M_{Client} and C_{Server} using various dynamic parameters, where these parameters do not include any detail about the identities of the sender or the receiver. Therefore, the adversary cannot obtain the identity of the sender or the receiver from the OTP. The adversary cannot link messages as the OTP is a variable string, which dynamically changes its content every time during communication. Thus, anonymity and unlinkability are preserved.
6) Resistance to offline password guessing attack: To correctly guess the OTP at M_{Client}, the adversary needs the PK_{cs}, which is not stored at M_{Client}. Likewise, to correctly guess the OTP at the C_{Server}, the adversary needs the PK_{mc}, which is not stored at the C_{Server}. Thus, the OTP of DRmcc resists the offline password guessing attack.
7) Immunity to session key retrieval attack: In the DRmcc protocol, the connection password and session key are the same, which is the OTP. The OTP is not exchanged between the M_{Client} and C_{Server} in plain text; rather it is encrypted at the sender side and decrypted at the receiver side using a one-time key, which consists of a

International Journal For Advanced Research In Science & Technology
A peer reviewed international journal www.ijarst.in
ISSN: 2457-0362

concatenation of P, G, and SSK values. Thus, DRmcc resists the session key retrieving attack, as the adversary does not have the encryption key to decrypt the OTP.

8) Resistance to asynchronization attack: With every connection, C_{Server} and M_{Client} simultaneously generates a new OTP. The new generated OTP is valid for a single connection only and is immediately used for that connection. Thus, DRmcc is resistant to asynchronization attack.
9) Immunity to shoulder surfing attack: In the RDmcc protocol, the OTP is not required to be keyed in for every new connection; rather, the C_{Server} and M_{Client} dynamically and automatically generates it. Thus, DRmcc is immune to the shoulder surfing attack.
10) Resistance to stolen smart card attack: DRmcc does not depend on the smart card; thus, it is not subjected to stolen smart card attacks. However, DRmcc as a dynamic protocol depends on the mobile device to generate the OTP for every connection. Retrieving the current OTP from the mobile device is protected with a biometric password of the mobile user such as fingerprint, eye print, or face recognition. Such passwords are difficult to crack and therefore, the DRmcc is also resistant to stolen device attack.
11) Mutual authentication: Both C_{Server} and M_{Client} are able to authenticate each other.
12) One-time password: For every single connection, a new password is created, which is valid only for a single connection.
13) Dynamic password generation: The utilized OTP is automatically generated and regularly updated by each M_{Client} and C_{Server} without human involvement.
14) Usability: DRmcc offers the benefit of being memory-wise effortless, easy to learn, and efficient to use.
15) Deployability: The DRmcc protocol does not require typing the password, and offers negligible-cost-per-user because it is lightweight in computation and communication. DRmcc also offers the benefit of not using a third-party for authenticating M_{Client} or C_{Server}.
16) Scalability: This is evaluated by measuring the complexity of operating the DRmcc protocol at each M_{Client} device and C_{Server}, particularly under limited memory resources and processor speed. In this study, the complexity is measured by calculating the computation overhead involved and extra communication cost.

V.EXPERIMENTAL RESULTS

This section presents experimental results that demonstrate the new features of DRmcc. Java simulation is developed to serve as a testbed for evaluating whether the proposed DRmcc protocol provides a mutual authentication communication. Experiments were conducted using a mobile user's cloud account and a mobile device that are registered to the C_{Server} using a unique IMSI. Table 1 shows information, which the M_{Client} uses to register with C_{Server} and the initial OTP generated at both M_{Client} and C_{Server}. The mobile user is prompted to enter the username and password. IMSI number is automatically extracted from the utilized mobile device and submitted along with the username and password to C_{Server}. The username, password, and IMSI are used at both M_{Client} and C_{Server} to generate the initial OTP.

TABLE 1
REGISTRATION INFORMATION

Registration Information	Mobile Client	Cloud Server
User ID	A9hme4da	A9hme4da
User Password	$xas7zbkm	$xas7zbkm
IMSI	545781549854164	545781549854164
Initial OTP	A9hme4da$xas7zbkm5 45781549854164	A9hme4da$xas7zbkm5 45781549854164

The initial OTP is then used to generate the new OTP. For any new connection, a new OTP is dynamically generated as a concatenation of the previous OTP, P, G, and SSK values. The values of P and G of the Diffie–Hellman algorithm in both the M_{Client} and C_{Server} are extracted from the previous OTP. The values of PRKs are randomly

International Journal For Advanced Research In Science & Technology
A peer reviewed international journal www.ijarst.in
ISSN: 2457-0362

generated for each M_{Client} and C_{Server}. Upon receiving the PRKs, each M_{Client} and C_{Server} computes its PK. The computed PKs are exchanged between the M_{Client} and C_{Server} to compute the SSK, which must be the same values at the M_{Client} and C_{Server}. For instance, values for Connection 1, as shown in Table 2, are 97 and 17 for P and G, respectively. PRK values for M_{Client} and C_{Server} are 845 and 512. PKs are 56 and 61 for M_{Client} and C_{Server}, respectively. The computed values of SSK are 35 at M_{Client} and C_{Server}.

Once verified by the server, M_{Client} gains access to the cloud service provider C_{Server} and the current OTP is replaced with the new one. However, the dynamic OTP password is not displayed to the mobile user, and it can be retrieved using biometric password of the mobile user. Table 2 shows the dynamic OTP, which is dynamically generated by M_{Client} and C_{Server} for every connection.

TABLE 2
DYNAMIC OTP GENERATED FOR EVERY CONNECTION

A: Dynamic OTP at M_{Client}

Connection	P	G	PRK	PK	SSK	OTP to be encrypted (32 Characters)
1	97	17	845	56	35	9hme4da$xas7zbkm5457 81549854164ed3d7
2	107	19	767	57	53	a$xas7zbkm54578154985 4164ed3d7105b51
3	107	23	312	53	89	zbkm545781549854164e d3d7105b51105b51
4	113	27	542	104	57	5781549854164ed3d7105 b51105b511148d5
5	127	29	293	97	16	9854164ed3d7105b51105 b511148d5136c54
6	109	29	764	73	78	4ed3d7105b51105b51114 8d5136c5410ad72
7	89	23	409	7	44	7105b51105b511148d513 6c5410ad72d9db8
8	97	27	198	64	64	51105b511148d5136c541 0ad72d9db8ed7dc
9	97	23	946	32	49	b511148d5136c5410ad72 d9db8ed7dced63d
10	89	19	547	24	29	48d5136c5410ad72d9db8 ed7dced63dd9c19

B: Dynamic OTP at C_{Server}

Connection	P	G	PRK	PK	SSK	Decrypted OTP
1	97	17	512	61	35	9hme4da$xas7zbkm5457 81549854164ed3d7
2	107	19	294	36	53	a$xas7zbkm54578154985 4164ed3d7105b51
3	107	23	654	99	89	zbkm545781549854164e d3d7105b51105b51
4	113	27	946	72	57	5781549854164ed3d7105 b51105b511148d5
5	127	29	594	8	16	9854164ed3d7105b51105 b511148d5136c54
6	109	29	277	104	78	4ed3d7105b51105b51114 8d5136c5410ad72
7	89	23	612	57	44	7105b51105b511148d513 6c5410ad72d9db8
8	97	27	497	27	64	51105b511148d5136c541 0ad72d9db8ed7dc
9	97	23	115	82	49	b511148d5136c5410ad72 d9db8ed7dced63d
10	89	19	707	6	29	48d5136c5410ad72d9db8 ed7dced63dd9c19

VI. RESULT EVALUATION

In this section, we evaluate the proposed DRmcc protocol by comparing its performance with other existing works [9][10][32] in terms of security, usability, deployability, computation overhead, and communication cost. To compare and rate relevant schemes across a common spectrum, we use the criteria suggested in [36], which analyzes the use of passwords in different authentication methods. Table 3 shows the short forms and symbols used for comparison purposes.

TABLE 3
SHORT FORMS AND SYMBOLS USED FOR COMPARISON PURPOSES.

Symbol	Description
+	Achieved the corresponding goal
–	Not achieved
$€_T$	Computation Overhead
$€_S$	Communication Cost
t_h	$€_T$ (one-time hash, h(.))
t_x	$€_T$ (one-time hex, hx(.))
t_e	$€_T$ (one-time encryption, E(.))
$t_{\bar{e}}$	$€_T$ (one-time decryption, D(.))
t_g	$€_T$ (key/PWD generation)

International Journal For Advanced Research In Science & Technology
A peer reviewed international journal www.ijarst.in
ISSN: 2457-0362

t_v	ϵ_T (verification process)
t_b	ϵ_T (load balancing process)
t_{me}	ϵ_T (exponentiation modulo)

In terms of security, the DRmcc protocol is evaluated based on the invulnerability to the known attacks listed in the adversary model. The known attacks involved in the comparative evaluation are MITM, playback, anonymity and unlinkability, offline password guessing, session-key retrieval, impersonation, asynchronization, shoulder surfing, and stolen smart card attacks. Table 4 shows that the DRmcc protocol is more efficient due to its immunity against the known attacks compared with other existing works.

In addition to its mutuality, dynamicity, and utility of OTP, Table 4 shows that the DRmcc protocol is more efficient than its most related works in terms of scalability. In this study, the complexity is measured by calculating the computation overhead involved and extra communication cost [37][38][39]. The scalability of the existing works is recorded as reported in the published papers. The scalability of the DRmcc protocol is evaluated by measuring the complexity of operating the DRmcc protocol at each M_{Client} and C_{Server}. RDmcc is more scalable due to the smaller number of processes involved in its algorithm as well as the fewer messages needed for the communication between M_{Client} and C_{Server}. For DRmcc, PK's message take 16 chars (128 bits) and OTP's message is 32 chars (256 bits). As shown in Table 4, compared with other schemes, the DRmcc protocol has the lowest computation overhead and requires the least communication cost as well.

In terms of usability, the DRmcc protocol offers the benefit of being memory-wise effortless, easy to learn, and efficient to use. As the user is not required to remember and input the password in the next authentication to access to the server, the protocol is effortless and efficient. Since DRmcc is proposed to be run on smartphones and servers without direct interaction from the users, it has the benefit of nothing-to-carry and physical effortless as the user is not required to carry any gadget other than a mobile device.

Finally, in terms of deployability, the DRmcc protocol is accessible as most disabled users can use a mobile phone without typing in the password. Based on the assumption that most users today own a mobile phone, DRmcc offers the benefit of negligible cost per user because it is lightweight in computation and communication. DRmcc also offers the benefit of not using a third party in the protocol and is unlinkable because the parameters used in generating the OTP are unique and specific to individual mobile devices. The mobile users must also provide/have access to their device to use the DRmcc; thus, this offers the benefit of requiring explicit consent in terms of security.

VII. CONCLUSION AND FUTURE WORKS

Reciprocal authentication is important to ensure that the communication between two parties is genuine. The DRmcc protocol attains the reciprocal authentication by using multifactor authentication, Diffie–Hellman key exchange, and onetime password. The SSK exchanged between the sender and receiver ensures a reciprocity of authentication between M_{Client} and C_{Server}. Multi-factor authentication with one-time password is feasible to prevent MITM attacks, especially replay attacks. Due to the OTP, which is valid for a single connection only and can be immediately used for that connection, DRmcc is resistant to asynchronization attacks. A unique feature of the DRmcc is its immunity to social engineering attacks, such as shoulder surfing, because the OTP is dynamically and automatically generated and does not need to be keyed in for every new connection. DRmcc is computationally less expensive. Thus, considering computational cost and robustness, the protocol can be

a good choice in authenticating and securing data communication in MCC environment. In addition to its security and efficiency as an authenticated protocol, DRmcc has various merits relevant to dynamicity, usability, and deployability.

In the future, the DRmcc protocol will be enhanced to secure the user credentials in case of physical loss of the mobile device. In particular, this enhancement will focus on securing the user credentials stored in the mobile database to prevent exposure to others. This task can be performed by developing a method that securely retrieves the current OTP from the mobile device using a biometric password of the mobile user. The biometric password, which can be used for this purpose, uses fingerprint, eye print, or face recognition. Such passwords are difficult to crack and therefore, the DRmcc is resistant to stolen device attacks.

ACKNOWLEDGEMENT

Funding support was provided by the fund of COMSTECHTWAS, joint research grant program for young scientists in OIC countries (No. 14-340 RG/ITC/AS_C).

REFERENCES

https://dora.dmu.ac.uk/bitstream/handle/2086/20398/DRmcc%20Accepted%20Draft_4_DMU.pdf?sequence=2&isAllowed=y

http://hdl.handle.net/2436/623412