A Dissertation Report

on

MULTI-FACTOR AUTHENTICATION SCHEME FOR ANTI-PHISHING USING MOBILE APP AND WEBCAM

Submitted in partial fulfillment of the
requirements for the award of degree of

Masters of Technology
In
COMPUTER SCIENCE AND ENGINEERING

By
SAJAL JINDAL
(17535023)

Under the guidance of
Dr. Manoj Misra
Professor

Department of Computer Science and Engineering
Indian Institute of Technology Roorkee
Roorkee – 247667

May, 2019

Candidate's Declaration

I declare that the work presented in this dissertation with title "**MULTI-FACTOR AUTHENTICATION SCHEME FOR ANTI-PHISHING USING MOBILE APP AND WEBCAM**" towards fulfilment of the requirement for the award of the degree of **Master of Technology** in **Computer Science and Engineering, Indian Institute of Technology Roorkee , India** is an authentic record of my own work carried out during the period of **June 2018 to May 2019** under the supervision of **Dr**. **Manoj Misra** , Professor, Department of Computer Science and Engineering, Indian Institute of Technology Roorkee, India. The content of this dissertation has not been submitted by me for the award of any other degree of this or any other institute.

Date: …….. (SAJAL JINDAL)

Place: Roorkee (17535023)

Certificate

This is to certify that the statement made by the candidate is correct to the best of my knowledge and belief.

Date: …….. Sign: ……..

Place: Roorkee Dr. Manoj Misra

Professor

Indian Institute of Technology Roorkee

ACKNOWLEDGEMENTS

I would first like to thank my thesis advisor Prof. Manoj Misra for guiding me throughout my thesis work, helping me whenever needed and being a constant source of motivation. I am really grateful for having such a wonderful and understanding mentor. I am also thankful to the Department of Computer Science and Engineering, IIT Roorkee for providing the valuable resources to aid my research. I would like to thank my fellow classmates and friends for being there and helping me overcome the obstacles in my thesis work.

Last but not the least, I would like to thank my parents and my brother for their blessings and support without which I would not have reached this stage of my life.

Sajal Jindal

ABSTRACT

Phishing attacks are one of the most serious threats faced by the users on the internet the attackers try to steal sensitive information such as login details, credit card details, etc. by deceiving the users to enter sensitive information on the phishing websites and thus leading to huge financial losses. These attacks involve using social engineering techniques to deceive the users. Many schemes have been proposed to detect phishing attacks but the amount of such attacks has not declined. New attacks like Active Man-In-The-Middle (MITM) phishing attacks have emerged which include Real Time Man-In-The-Middle (RT MITM) and Controlled Relay Man-In-The-Middle (CR MITM) phishing attacks. These attacks allow the attackers to obtain the users' account details and relay them in real-time. Similarly, the attacker can lure the user to enter details on a spoofed app and thus gain access to the user's account. The existing popular authentication schemes fail to address these attacks. Therefore, there is a need to prevent phishing attacks such as active MITM phishing attacks, app spoofing and malicious browser extension based attacks by creating an anti-phishing user authentication scheme. In this thesis, we propose a novel user authentication scheme which enables the user to log into his/her account without memorizing any password or any other authentication token. The proposed authentication scheme requires the user has to scan a dynamically generated QR-code using the smartphone app and then verify the image captured by the webcam and sent on the smartphone via push notification. Thus, the complete authentication procedure requires minimal user involvement and implements automatically. We have implemented and evaluated the proposed scheme in terms of usability, deployability and security parameters and the results depict that the proposed authentication scheme performs well and can be used as a secure user authentication scheme.

TABLE OF CONTENTS

List of Figures

List of Tables

CHAPTER 1

INTRODUCTION

1.1 Introduction

Phishing is a deception technique which is used to steal users' credentials with the goal of obtaining their personal information [1]. Phishing is an attack scenario in which the attackers known as the phishers masquerade as authentic website and somehow lure the users into entering their credentials. The term 'phishing' was first used in 1996 when the phishers stole credentials of American Online (AOL) users [2]. The phishers use various social engineering mechanisms for attacking the users, either by making them enter their details on the phishing website or via reply to phishing emails. The URL link of the phishing website is generally spread by spreading email in bulk or other communication medium. Naïve users, who still don't verify the domain names or other technical information tend to follow the instructions mentioned on the phishing website or email. In this way, the users generally reveal their credentials which are used by phishers for various malicious purposes such as identity theft, online credit card, and banking frauds, etc. [2-6]. For redirecting users to the phishing website, the attackers can use one of the following techniques- email spoofing, deceptive links, malicious browser extensions, etc. There are various kinds of attacks that are used by the attackers to access the user's account, such as- Real-Time MITM, Controlled Relay MITM, malicious browser extension based phishing attacks.

The single-factor authentication schemes are vulnerable to traditional phishing attacks and thus multifactor authentication schemes were proposed. Multi-factor authentication is a way of combining two or more authentication factors so as to add another layer of security to the system. The authentication factors that can be chosen for multi-factor authentication are- *something-you-know*, *something-you-have* and *someone-you-are.*

The problem of phishing can be solved by educating the users about the phishing attacks, using phishing detection schemes to determine whether the website is authentic or by using phishing

prevention schemes that can thwart phishing attacks. As, the phishing attacks are becoming more and more sophisticated, there is a race between the attackers creating new attacks and the researchers proposing new phishing prevention schemes. Thus, there is a need for a web-authentication scheme that can thwart phishing attacks.

After the first official cyber threat was recorded in 1996, many anti-phishing organizations such as APWG, RSA [7], Phishtank, etc. have not only recorded numerous phishing attacks but also analyzed them. According to the APWG phishing trends report [8], the number of phishing websites recorded in the 4th quarter of 2018 were 138,328 which is not a significant decrease from previous quarter. Thus, we can see that even after the proposal of various phishing detection and prevention schemes, the number of phishing websites and the number of phishing emails do not decrease drastically. Table 1.1 shows the phishing attack trends as per APWG report [8, 9] from H1 2017 to Q4 2018. The parameters considered by the reports for recording and analyzing the attack trends are number of unique phishing websites detected, number of unique phishing emails reported and number of brands targeted. The most targeted industry sector by the attackers have been payment with 33% in 4th quarter of 2018 followed by webmail or software as a service (SaaS) with 20% in 4th quarter of 2018.

Table 1.1 Current Phishing Attack Trends [8, 9]

Parameter	**4Q2018**	**3Q2018**	**2Q2018**	**1Q2018**	**4Q2017**	**3Q2017**	**1H2017**
Number of unique phishing websites detected	138,328	151,014	233,040	263,538	180,757	190,942	291,096
Number of unique phishing email reports received by APWG from consumers	239,910	270,557	264,483	262,704	233,613	296,208	592,335
Number of brands targeted by phishing campaigns	836	777	786	746	939	915	2660
Most Targeted Industry Sectors (Payment)	33.0%	38.2%	36%	39.4%	42%	41.99%	45%

1.2 Motivation

There has been continuous efforts in the past by the researchers to provide an authentication scheme which is secure as well as usable. Multi-factor authentication schemes were proposed because single factor authentication schemes were unable to handle traditional phishing attacks.

Multi-factor authentication schemes such as OTP based, QR-code, push login and graphical password based authentication schemes are not secure against advanced phishing attacks including RT MITM (Real Time Man-In-The-Middle) phishing attack and CR MITM (Controlled Relay Man-In-The-Middle) phishing attack. The phisher gets access to the user's account in OTP based authentication scheme by deceiving the user to enter the login details on the phishing website and the entered information is then relayed by automated means on the authentic website. Similarly QR-code based protocols can also be attacked by relaying the server generated QR-code on the phishing website. When the relayed QR-code is scanned by the user, then the authentication process is complete and the server send the user's account to the attacker's browser. A detailed explanation of RT MITM phishing attack on OTP and QR-code based authentication protocol has been explained in section 2.3. The attacker can compromise CAPTCHA based graphical password based schemes either by monitoring the user's desktop screen or by relaying the desktop terminal over the user's terminal. Since biometric based authentication schemes do not offer 100% accuracy and require an external hardware, thus it is not user-friendly. Moreover, the attacker can spoof user's biometrics such as fingerprint, iris, voice, etc. Though the hardware token based schemes provide extra security than other authentication schemes but they lack in usability as the user has to carry additional hardware token for the purpose of authentication. In addition to this, the hardware token schemes that also require the user to enter any security key are not safe as the attacker can obtain the security key via malicious browser extensions or reverse engineering.

D. Wang et al. [44] has mentioned that a number of authentication schemes have been proposed but very few of them are able to actually provide security. An attacker can compromise the schemes by either relaying the credentials from the phishing website to the authentic website or by capturing the user's remote desktop or by installing spoofed mobile application on the user's smartphone. Hence, there is a need of an authentication scheme that is able to handle all these attacks ranging from traditional phishing attacks to advanced phishing attacks which include RT

MITM, CR MITM. In addition to this, the authentication scheme should also be able to handle app spoofing and attacks caused due to malicious browser extensions. Hence, a secure authentication scheme is required that not only uses a token that cannot be spoofed or obtained, but also provides usability and deployability.

1.3 Problem Statement

The main objective of this thesis is as follows:

"To propose a new reliable authentication scheme that can thwart traditional and advanced phishing attacks."

The above problem statement can be divided into smaller objectives as follows:

1 Identifying and simulating the attack scenarios through which the attackers are able to steal user's credentials.

2 To design, implement and validate a secure user authentication scheme that can thwart RT MITM, CR MITM phishing and malicious browser extension based phishing attacks in addition to traditional phishing attacks.

3 The scheme proposed should not only be able to handle traditional and advanced phishing attacks, but the scheme should also be user-friendly as well as deployable so that it does not require any major changes and can be easily used for daily-login activities.

1.4 Contribution of Thesis

The foremost contribution of the thesis is to determine the security vulnerabilities of the existing authentication schemes. The existing multi-factor authentication schemes cannot withstand Active Man-In-The-Middle (MITM) attack and malicious browser extension based attacks. Thus, there is a need for a secure user authentication scheme that can address these attacks.

This thesis propose a novel secure user authentication scheme which makes use of the user's smartphone and the desktop's webcam. The scheme requires the user to scan the QR-code displayed on the browser using the smartphone app which stores a secret key for that particular user in a secure storage. The webcam then captures user's image, adds a random text at random position in the image and sends it to the user for verification. Once the user verifies the authenticity

of the image, the user gets access to the account. Thus, the scheme proposed in this thesis requires minimal user involvement as there is no need for the user to remember any username or password. The secret key is shared between the smartphone app and the web server, where it is kept in a secure storage. Since the user does not enter any information, there are less chances for the attacker to retrieve any user information using the phishing website.

In this thesis, we have also evaluated the performance of the proposed scheme with respect to time, resources utilized and also compared it with other existing schemes on the basis of usability, deployability and security. We have also shown that the proposed scheme is user-friendly with the help of a user survey. The results of all the experiments infer that the proposed scheme is secure, user-friendly and can be easily deployed without any major changes on the server as well as the client side.

1.5 Organization of Thesis

This chapter gives a brief introduction to phishing and its affect in various industry sectors. It also describes our problem statement and our motivation behind it. It concludes with the contribution of this thesis.

Chapter 2 gives a brief description about traditional as well as advanced phishing attacks. It also gives a brief description about various multi-factor authentication schemes, their advantages and the research gaps present in them.

In Chapter 3, a detailed design of the registration and the login phase has been explained. This chapter also discusses the overall workflow and implementation of the proposed authentication protocol.

Chapter 4 is dedicated to the performance evaluation of the proposed user authentication scheme in terms of time required for registration and login and the resources utilized by the proposed protocol. It also shows the comparison of the proposed protocol with the existing authentication protocols on the basis of usability, security and deployability.

Chapter 5 concludes our work and describes how it can be taken forward.

CHAPTER 2

BACKGROUND AND RELATED WORK

2.1 Phishing Attacks

Phishing attacks can be categorized into traditional as well as advanced phishing attacks.

2.1.1 Traditional Phishing Attacks

Attackers use a variety of techniques to deceive the user [10, 11], so as to gain access to his/her account details. These techniques include email spoofing, website spoofing and exploitation of browser vulnerabilities. The phishers also use link manipulation, in which the URL of the authentic website or words are used to hide the actual phishing URL. As a result, the user will open the phishing website considering it as authentic. The attacker can also use web scripting language like Java Script to customize the browser on the user's desktop. The attacker can hide the address bar or show an authentic URL in the address bar or use images having phishing information instead of using text as many phishing detection filters cannot detect images [12]. The user may also be asked to enter their credentials by using a pop-up window. The phishers can also inject malicious content into websites [13]. The attacker can also direct the users to the phishing websites by changing the DNS.

The traditional phishing attacks can easily be launched as there are a variety of phishing toolkits and simulator using which, the attackers can easily initiate the attack. With the help of these toolkits and simulators, even a beginner in the field of hacking would be able to initiate the attack easily because the simulators makes an exact copy of the desired website, which the attacker can run on a server so that the user can access it on the internet. These toolkits also enable the attackers to send the spoofed emails, containing the phishing website URL, to the users. The users get easily deceived and lured to open the phishing website URL asking for their credentials. The traditional phishing attacks are able to break conventional one-way user authentication schemes but not the latest two-factor authentication schemes.

2.1.2 Advanced Phishing Attacks

1. Real-Time Man in the Middle (RT MITM) Phishing Attack

RT MITM [16, 17, 20] phishing attack is a sophisticated attack in which the attacker place themselves between the user or the client and the server. The phishers direct the users to the phishing website which looks similar to the authentic website. The user is then asked to enter their personal information which is stored by the attacker. This information is relayed by the attacker to the authentic webpage in real-time. This process of relaying the information can be done manually or by automated means to speed-up the attack. The server on receiving the information from the attacker, will verify it and then send the user's account on the attacker's browser. Thus, the attacker is able to gain access to the user's account by coming in the path between the client and the server. Google 2-step verification [15] and WhatsApp Web [33] authentication are vulnerable to RT MITM phishing attack. A detailed simulation of RT MITM attack on these schemes has been shown in section 2.3.

2. Controlled Relay Man in the Middle (CR MITM) Phishing Attack

In CR MITM [16, 17, 20] phishing attack, a phisher casts his/her screen over the user's desktop using popular applications like TeamViewer. Thus, the attacker simply relays his/her desktop screen over the client's terminal. The attacker then opens the authentic website which is shown to the user. The user gets deceived when he/she sees the screen and gets lured to enter the credentials. The server verifies the credentials and send the account information to the attacker's terminal whose screen was relayed over the user's desktop. CR MITM phishing attack makes one-way as well as two-way authentication schemes vulnerable.

3. Malicious Extension Based Phishing Attack

The malicious browser extensions [30, 31, 32] can also be used by the attackers to perform phishing attack so as to obtain the user's credentials. The malicious browser extensions ask for permission from the user to provide some functionality to the users in the foreground. Now, the same set of permissions are used to carry out an attack in the background so as to obtain user's credentials. An example of malicious browser extension would be an extension that checks and eliminates grammar errors for the user in the foreground. This extension can get the permission to access the website contents in the browser and also the URL in the address bar. With the help of

these permissions, the attackers have enough power to initiate a phishing attack for stealing the user's credentials in the background. There are various attacks that can be performed by the malicious browser extensions such as keylogging, screen logging or password/data sniffing. The malicious browser extensions can steal the information even before it is encrypted. Therefore, the password manager based schemes are also vulnerable to malicious browser extensions.

2.2 Existing Multi-factor Authentication Schemes

There are various web authentication schemes available using which the user can login to the website. The simplest authentication scheme is by providing username and password. The existing authentication schemes can be categorized as:

2.2.1 Authentication using OTP/PIN:

The OTP based schemes such as Google 2-step [15] requires the user to first enter his/her username and password. The server then generates and sends an OTP (one-time password) to the user's registered phone via SMS. If the credentials and the OTP is entered correctly by the user, then the user will be logged into the website. Other OTP/PIN based two-factor authentication scheme is SAASPASS [14] in which the SAASPASS application is installed by the user on his/her smartphone which is linked to the user's personal web account. At the time of login, SAASPASS generates and sends a 6-character PIN to the server as well as the user. This 6-character PIN is updated and sent every 30 seconds. Both the above mentioned schemes are vulnerable to MITM phishing attacks as the attacker can get the OTP/PIN with the help of a phishing website or through malicious browser extension.

2.2.2 Authentication using graphical password and CAPTCHA:

In Leung et al.'s [20] scheme, the concept of flash based OTP CAPTCHA was proposed to provide security against the MITM and malicious browser extension based attacks in which the user's screen is captured by the attacker to steal credentials. The user enters the username and mouse click coordinates of the OTP CAPTCHA which consists of moving letters and numbers. The user's mouse click coordinates on the OTP plugin and the click timestamp is also taken as an input. Zhu et al. [21] proposed a scheme in which CAPTCHA was used as graphical password. The user enters the password by clicking characters in the CAPTCHA and the click coordinates are used by

the server for verification. Both the above mentioned authentication schemes are not secure as Leung et al.'s [20] scheme is vulnerable to CR MITM attack and Zhu et al.'s scheme [21] can be compromised using MITM and screen logging attacks.

2.2.3 Authentication using push notification:

Push notification based login schemes are provided by Yahoo mail [22, 45], in which the user enters his/her username on the website. The server verifies the user's credentials and sends a push notification message to the registered mobile app. The user gets access once he/she verifies the push notification message received on the mobile app. The push notification based login schemes are also provided by Google. The push notification based authentication schemes cannot withstand MITM attacks.

2.2.4 Authentication using password managers:

Password managers [23] helps the user by storing their credentials for different websites and they automatically fill these credentials when the user logs into that website. Most of the password managers store the user credentials in browser storage. Ross et al. [24] proposed a similar scheme in which, the browser extension is used to modify the password entered by the user with the help of the SALT stored at the client machine and the domain information of the website. Though the attacker won't gain anything even if the credentials are entered by the user on the phishing website, yet this scheme is not secure against malware attacks as well as it is client dependent because the salt is stored in plaintext on the client machines.

2.2.5 Authentication using hardware token:

This category of authentication scheme requires the user to carry a special hardware token such as USB, security keys, smart cards, etc. for authentication. These hardware tokens may store some passwords or cryptographic keys which are communicated during the authentication process. Tricipher scheme [25] uses multipart credentials where one part remains with the user and the other part of the credentials is stored in a secure appliance kept in the enterprise data center. A secret key is also stored in the user's device which is also known to the server. The username and password entered by the user is encrypted using this secret key. The secure appliance encrypts the user's credentials using the credentials stored on it, which is then sent to the server and thus the

authentication process is completed. Other hardware token based authentication schemes are RSA SecurID which updates the authentication code every 60 seconds as well as U2F security keys such as Yubikey [26] which follows the U2F protocol for user verification. In RSA SecurID, an authentication code is usually generated every 60 seconds using the clock and random seed which is provided to the token by the RSA server at the time of purchase of the device. The server computes the authentication code using the seed stored in its database for the token and the clock and verifies it with the code given during the login process.

2.2.6 Authentication using QR-code:

Xie et al [16] proposed a QR-code based authentication scheme in which, the user first gives the username and password to the web-browser. The server validates user's credentials and renders a barcode on the desktop screen which is scanned by the users using mobile app and a vouch request is generated in the form of a barcode which is scanned by the PC webcam. This authentication scheme claims to be secure against MITM phishing attack and Diffie-Hellman algorithm is used for securing the communication channel between the browser and the server. Kim et al. [17] also proposed a QR-code based authentication scheme in which the IP address of the smartphone was used to verify that the user and PC are in proximity. In Mukhopadhyay et al.'s scheme [18], a third-party verifier is used for checking the user's credentials and after verification, sends a challenge in the form of a QR-code to the user. The user, in turn, scans the QR-code using the mobile app and sends back an encrypted response to the third-party verifier. In Dodson et al.'s [19] scheme, the server sends a QR-code consisting of server challenge to the user. The user then scans the QR-code using his/her smartphone and a challenge response is sent to the server. Once the server verifies the challenge response, the user can access his/her account.

Ritwik et al. [46] also proposed a QR-code based authentication scheme in which the user first enters the credentials and then the browser extension establish a two-way authentication channel between the user and the browser. The user requests for the token via browser extension and the server generates the token after verifying the user's token request. The webpage then asks for username, password and token generated by the server. The browser extension gives the token to the webpage only if the domain of the original token request is same as that of the current webpage.

We evaluated the performance of the scheme proposed by Ritwik et al. [46] on the basis of usability, security and deployability which has been summarized in Table 2.1.

Table 2.1 Comparative Analysis of Ritwik et al. [46]

			Google 2-Step [15]	SAASPASS [14]	Xie et al. [16]	Kim et al. [17]	Mukhopadhyay et al. [18]	Dodson et al. [19]	Leung et al. [20]	Zhu et al. [21]	Tricipher et al. [25]	RSA SecurID token [27]	Yubikey U2F [26]	Push Login [22]	Password Managers [23, 24]	U-PWD [28, 44]	Ritwik et al. [46]
Usability	1	Memory wise effortless	×	×	×	●	×	✓	×	×	×	×	×	●	●	×	●
	2	Scalability for users	×	×	×	×	×	✓	●	●	×	×	×	✓	✓	×	●
	3	Nothing to carry	●	●	●	●	●	●	✓	✓	×	×	×	●	✓	✓	●
	4	Physically effortless	×	×	×	×	×	✓	×	×	×	×	×	●	✓	●	✓
	5	Easy to learn	✓	✓	●	✓	✓	✓	●	●	✓	✓	✓	✓	✓	✓	✓
	6	Efficient to use	●	●	×	●	●	✓	×	×	✓	✓	✓	✓	✓	✓	✓
	7	Infrequent errors	●	✓	×	×	●	●	×	●	✓	✓	✓	✓	✓	✓	✓
	8	Easy recovery from loss	●	●	●	●	●	●	✓	✓	●	●	●	●	●	●	●
Deployability	9	Accessible	●	●	×	×	×	×	×	×	✓	×	✓	●	✓	✓	●
	10	Negligible cost/user	×	●	●	●	●	●	×	✓	✓	×	×	●	✓	✓	●
	11	Server compatible	×	×	×	×	×	×	×	×	×	×	×	×	✓	✓	×
	12	Browser compatible	✓	✓	●	●	●	●	●	●	✓	✓	●	✓	✓	✓	×
	13	Mature	✓	✓	×	×	●	●	×	×	●	✓	✓	✓	✓	✓	●
	14	Non-Proprietary	×	●	✓	✓	✓	✓	✓	✓	×	×	×	●	✓	✓	✓
Security	15	Resilient to physical observation	●	●	✓	✓	✓	✓	●	●	●	✓	✓	●	●	×	✓
	16	Resilient to target impersonation	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	×	×	✓
	17	Resilient to throttled guessing	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	×	×	✓
	18	Resilient to unthrottled guessing	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	×	×	✓
	19	Resilient to internal observation	×	×	×	●	×	×	●	●	×	×	●	●	×	×	✓
	20	Resilient to leak from other verifiers	✓	●	✓	✓	●	✓	✓	✓	●	●	●	✓	●	×	✓
	21	Resilient to Phishing	×	×	×	×	×	×	×	×	✓	×	✓	×	✓	×	✓
	22	Resilient to Theft	✓	✓	✓	✓	✓	×	✓	✓	✓	✓	✓	✓	×	●	✓
	23	No Trusted Third Party	✓	×	✓	✓	×	✓	✓	✓	×	×	×	✓	×	✓	✓
	24	Requiring explicit consent	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	×	✓	✓
	25	Unlinkable	✓	✓	✓	●	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		Benefit Offered Count	**11**	**10**	**10**	**10**	**9**	**14**	**11**	**12**	**13**	**12**	**13**	**14**	**14**	**13**	**16**

2.3 Simulation of Attack Scenarios

1. An attack scenario to break Google 2-step [15] using RT MITM phishing attack has been implemented. In this attack scenario, the URL of the phishing website is sent to the user via email or other communication medium. The user opens the phishing website and he/she is lured to enter the credentials on the phishing website which appears same as the original google accounts website. The credentials entered on the phishing website are stored on the web hosting server and from there, the attacker can relay these credentials on the authentic website in real-time. The authentic website, now, requests the user to enter the OTP sent to the registered mobile. A similar webpage is shown to the user, where the user will enter the

OTP, which in turn will be relayed by the attacker on the authentic website and thus the attacker will be logged into the user's account.

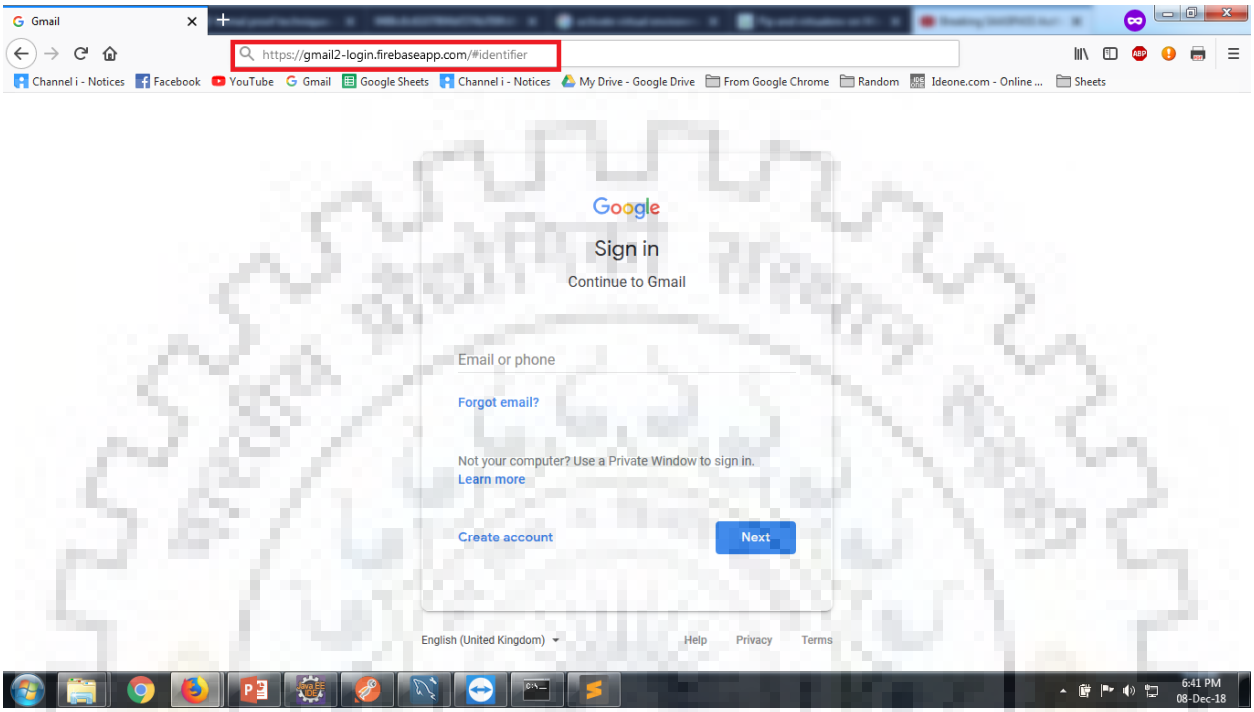


Fig. 2.1 Phishing website of the Google 2-step [15] authentication scheme

2. Attacking QR-code based scheme such as WhatsApp Web [33] has been implemented in which the attacker relays the QR-code from the authentic website to the phishing website in real-time. The QR-code gets updated after every 30 seconds and thus the phishing website will also show the updated QR-code in real time. The link of the phishing website on which the QR-code has been relayed, is again sent to the user. Now, the attacker opens the authentic website of WhatsApp Web and relays the displayed QR-code on the phishing website. When the user scans the QR-code using his/her smartphone, then the user information will be sent along with session token to the server. The server will verify the received information and send the user's account details on the browser which is open on the attacker's desktop. This way the attacker will get access to the user's account.

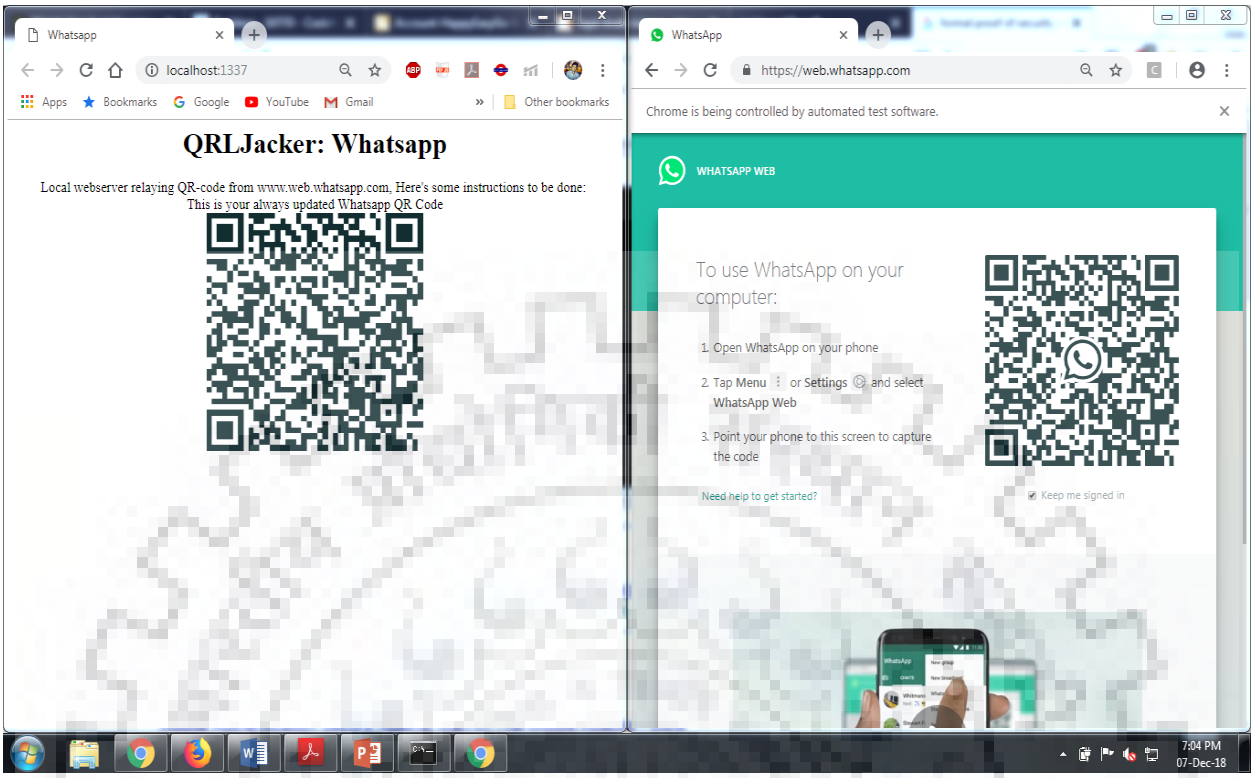


Fig. 2.2 QR-Code getting relayed from the authentic website to the phishing website(QrlJacking)

2.4 Research Gaps

1. The OTP/PIN based authentication schemes [14, 15] are vulnerable to RT MITM phishing as the attackers can deceive the users by showing them the phishing website and using the credentials entered by the user, getting access to the user's account in real time. An attack scenario has been implemented to show that the OTP/PIN based authentication schemes are vulnerable to RT MITM phishing attack.
2. QR-code based schemes are also not secure as they can be attacked using MITM phishing attacks. An attack scenario has been implemented in which the QR-code is relayed from the original website to the phishing website in real-time to hack the target's WhatsApp web account. This process of cloning the QR-code on the phishing website is known as QRLJacking [29].
3. QR-based schemes discussed in section 2.2.2 are vulnerable to various kind of attacks. Xie et al's [16] scheme can be compromised by the attacker using a spoofed malicious browser extension to obtain the username, password and establish an authentic session in real-time using the authentic website extension. Mukhopadhyay et al's [18] scheme is vulnerable to

malicious extension based phishing attacks as the username and password can be spoofed and when the victim will scan the QR-code using the smartphone app, the attacker will get access to the user's account. Dodson et al's scheme [19] can be attacked using MITM as the QR-code can be relayed to the victim's screen in real-time and scanning the QR-code using the user's smartphone will authenticate the attacker's browser's session.

4. Graphical password based authentication schemes [20, 21] are not able to stand against CR MITM attacks. Using the malicious browser extensions, the attacker can obtain the mouse-click information and then predict the graphical input.
5. Push notification based login schemes [22, 45] are not secure against RT MITM phishing attacks because the username can be obtained using the phishing website and when the server will send the push notification message, the user will approve the push notification message received in deception.
6. Password managers [23] can be easily attacked by the malicious browser extensions as the information auto-filled can be sniffed.
7. Hardware token schemes require the user to carry an extra security key or token.

CHAPTER 3

MULTI-FACTOR AUTHENTICATION SCHEME FOR ANTI-PHISHING

3.1 Proposed Scheme

The proposed multi-factor authentication scheme uses a trusted mobile application and a webcam on the client's machine (desktop/laptop). Whenever the user wants to access his/her account on the website, then a QR-code will be displayed on the webpage by the server. This authentication protocol assumes that the mobile application in user's phone is trusted. The user scans the QR-code displayed on the webpage using the mobile application. The application then sends the user's data along with the session token obtained from the QR-code to the server. The server then gives a prompt on the client's machine to access the webcam and captures user's image using the webcam. The server then adds a random text at random position in the captured image and send it to the smartphone application via push-notification for user's approval. The user will then approve or reject the authenticity of the received picture. The user will get access to the account once he/she verifies the authenticity of the image and approves it. Thus, the server authentication is done with the help of the picture taken on the legitimate user's client machine because only the legitimate website can take user's picture using webcam and send the same picture to the user's mobile phone.

3.1.1 Assumptions

1. User's desktop is assumed to have a webcam which is cheap and easily available.
2. It is assumed that the smartphone app used during the registration is authentic and like many other schemes, the new user registration is secure and free from attacks.
3. The proposed scheme assumes that HTTPS communication channel has been used to transfer the date between client and server. Moreover it is secure from network sniffing and can be used to exchange secret keys.
4. Authentic website servers and the information stored in their databases are assumed to be secure.

3.1.2 Threats

a) **MITM Phishing:** The attacker can lure the user by directing him/her to the phishing website and then obtain personal information. The attacker can either relay this information in real-time (RT MITM) or can relay his/her desktop screen on the remote user's terminal (CR MITM) using applications such as TeamViewer to steal credentials.

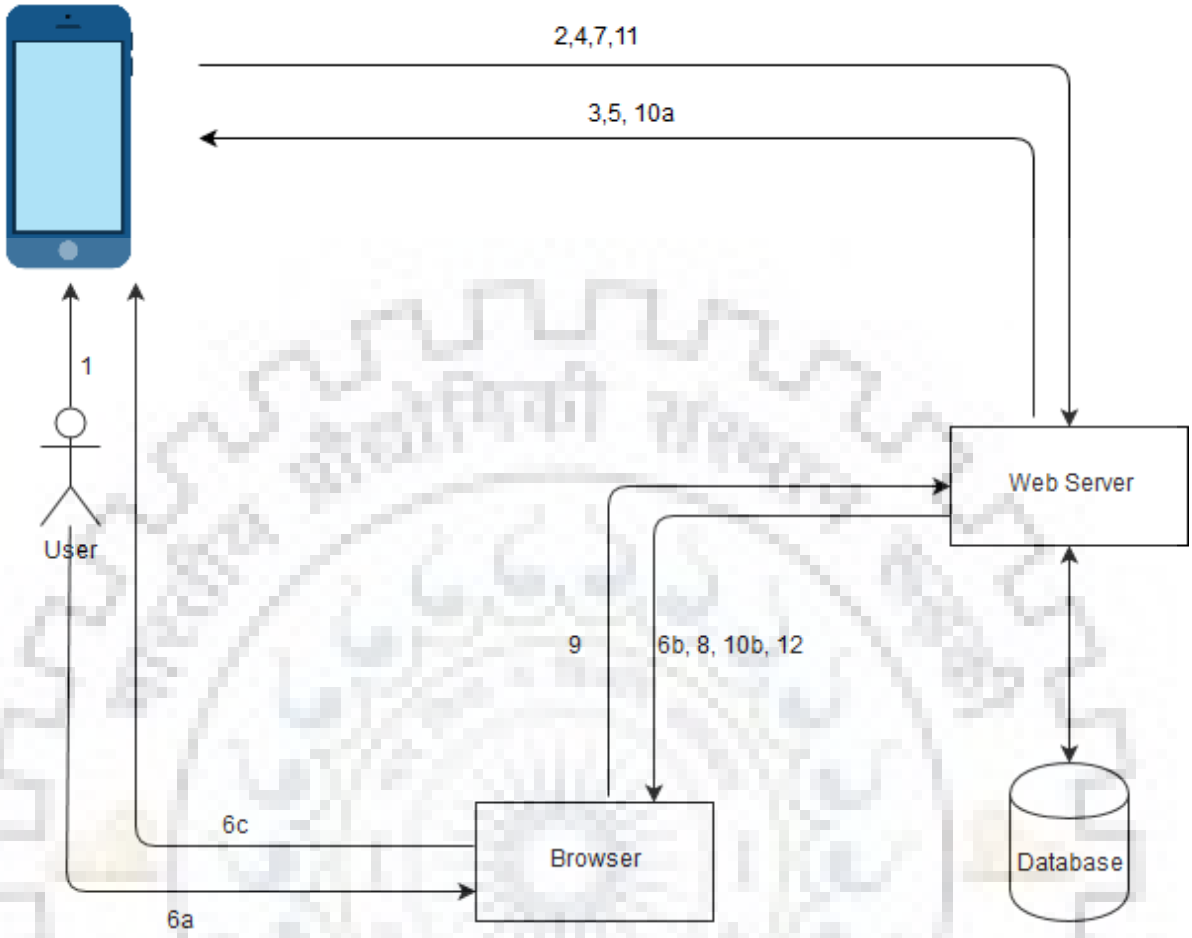
b) **Malicious browser extension based phishing attacks:** The malicious browser extensions ask for permission from the user to provide some functionality to the users in the foreground. Now, the same set of permissions are used to carry out an attack in the background so as to obtain user's credentials. The malicious browser extensions can perform keylogging, screen logging or password sniffing in the background.

c) **App spoofing:** The attacker can create a spoofed Android App [34] which looks similar to the authentic app required for login. The attacker can then install this spoofed app on the user's smartphone and lure the user into entering his/her credentials over this app.

3.1.3 Overall Workflow

Figure 3.1 shows the overall workflow for the proposed scheme which has the following steps:

1. User opens the smartphone app and enters the username, phone number, email and password for registration.
2. The smartphone app sends user's registration details to the web server for verification.
3. The web server then validates the registration details and sends an OTP to the user's registered phone.
4. User enters the OTP on the smartphone app which is sent to the web server. The server verifies the OTP and store details of the user in the database.
5. Meanwhile, the web server generates a secret key and shares it with the smartphone app where it is stored in a secure storage.
6.
    a) User opens the website on the browser.
    b) Web server renders a dynamically generated QR-code on the screen, which contains the session token.
    c) User scans the QR-code using the smartphone app to fetch the session token.



Note: Steps 1 to 5 show the registration procedure and 6a to 12 show login procedure.

Fig.3.1 Overall workflow for the proposed scheme

7. The smartphone app generates a request data (RD) consisting of the user-id, timestamp and session token and encrypts it using secret key. The encrypted request data is then sent by the app to the server.
8. The web server decrypts the encrypted request data using the shared secret key and verifies the token. The server then prompts the browser to access the webcam of the client's machine.
9. The webcam captures the user's image, embeds a random text at random position over the captured image and sends it to the server.
10.
    a) The image taken by the webcam is sent by the web server to the smartphone via push notification.
    b) Meanwhile, the web server also sends the same image on the browser.

11. The user verifies the image and the text embedded over it with the image on the browser and sends a response (approve/reject) to the server.
12. The login process is complete and the server sends the user's account details to the browser.

3.1.4 Registration

The registration phase involves registration of the user on the mobile app. Thus, the entities that are involved in the registration phase are – the mobile app and the web server. The user registration is done on the mobile app which will store the details of the user that will be used at the time of login. The user registration in the proposed authentication protocol has been shown in the figure 3.2. The steps for registering the user are as follows:

a) The user will first enter relevant details in the mobile app such as username (U_{ID}), password (PWD), email address (Email-ID), phone number, etc.
b) The details given by the user are then sent to the web server over HTTPS session. The web server generates an OTP (one-time password) and sends the generated OTP to the phone number given by the user.
c) The user then enters the OTP received on his phone which the web server verifies.
d) The web server then generates a hashed password (hpwd) from the password (pwd) entered by the user in step (a) and a random salt ($salt_1$).
e) The web server also generates a secret S using password based key derivation function 2 (PBKDF2). The parameters that are used in the PBKDF2 function are the password (pwd), a random salt ($salt_2$), iterations (iters) and key length (keylen).
f) The web server then generates another random salt ($salt_3$) which is used for encrypting the secret (S) generated in the above step.
g) After the user presses the OK button on the mobile screen, a confirmation message is sent to the web server over HTTPS session and all the user details along with the shared secret (S) is stored in the web server as well as the mobile database which is secured by the Android Keystore API [35].
h) The web server finally sends a message acknowledging the successful registration of the user.

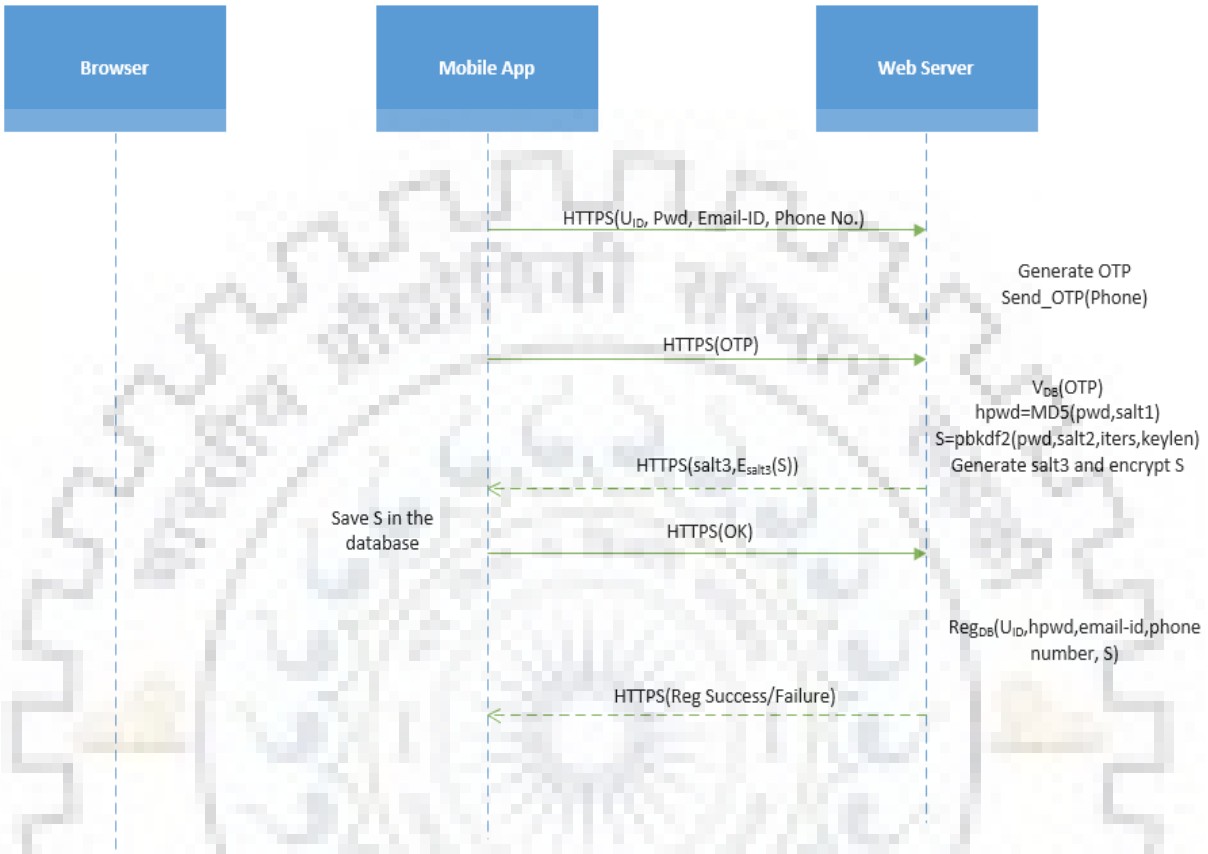


Fig.3.2 Mobile Registration Phase

3.1.5 Login

The proposed authentication protocol makes use of 3 entities for logging the user into the website. The entities used are – the mobile app, browser and web server. The user will also need a webcam for logging into the system. To login into the website account, the user must be logged into the mobile app of the website. The reason is that the push notification is sent to the app on which the user is logged in. For logging into the mobile app, the user enters the user-id (UID) and password (pwd), which is sent to the web server for verification. Since most of the users do not logout from their account, the chances of re-login to the mobile app is less. The login procedure is as follows:

a) The user first opens the login webpage on the browser.
b) The website login page displays a QR-code which contains session token generated by the web server.
c) The user scans the QR-code rendered by the website using his mobile's camera.
d) After the QR-code is scanned by the mobile app, the session token is retrieved from the QR-code. Now a secret (S') is generated by using SHA-256. The parameters used are the secret (S) stored in the Android Keystore API [35], user-id (UID) and the timestamp (TS).
e) The mobile app also generates request data (RD) which consists of user-id (UID), timestamp (TS) and session token obtained from the QR-code. The request data is then encrypted to form ERD using the secret (S') calculated in step (d) and sent to the web server along with user-id (UID), timestamp (TS) over HTTPS session.
f) The web server also generates a secret (S") similar to that in step (d) using the secret key (S) stored in the server's database, received user-id (UID) and timestamp (TS).
g) The encrypted request data (ERD) is then decrypted using the secret (S") generated in step (f). The web server then verifies the user-id (UID), session token and the timestamp (TS) obtained from the decrypted request data (RD).
h) A pop-up window then comes up on the browser asking permission to access the client's machine webcam. The user must allow access so as to login to the website.
i) The webcam then captures an image of the user, adds a random text at a random position over the image taken and sends it to the web server, which in turn forwards this picture to the user's mobile app via push notification.
j) The smartphone app shows the image received via push notification to the user asking for login approval.
k) At this step, the user will verify the image received and will approve login attempt if and only if the login attempt is done by the user himself/herself as well as the image received is authentic.
l) The mobile app sends the user's response (Approval/Reject) to the web server and based on the response, the web server shows user his/her account or displays login error.

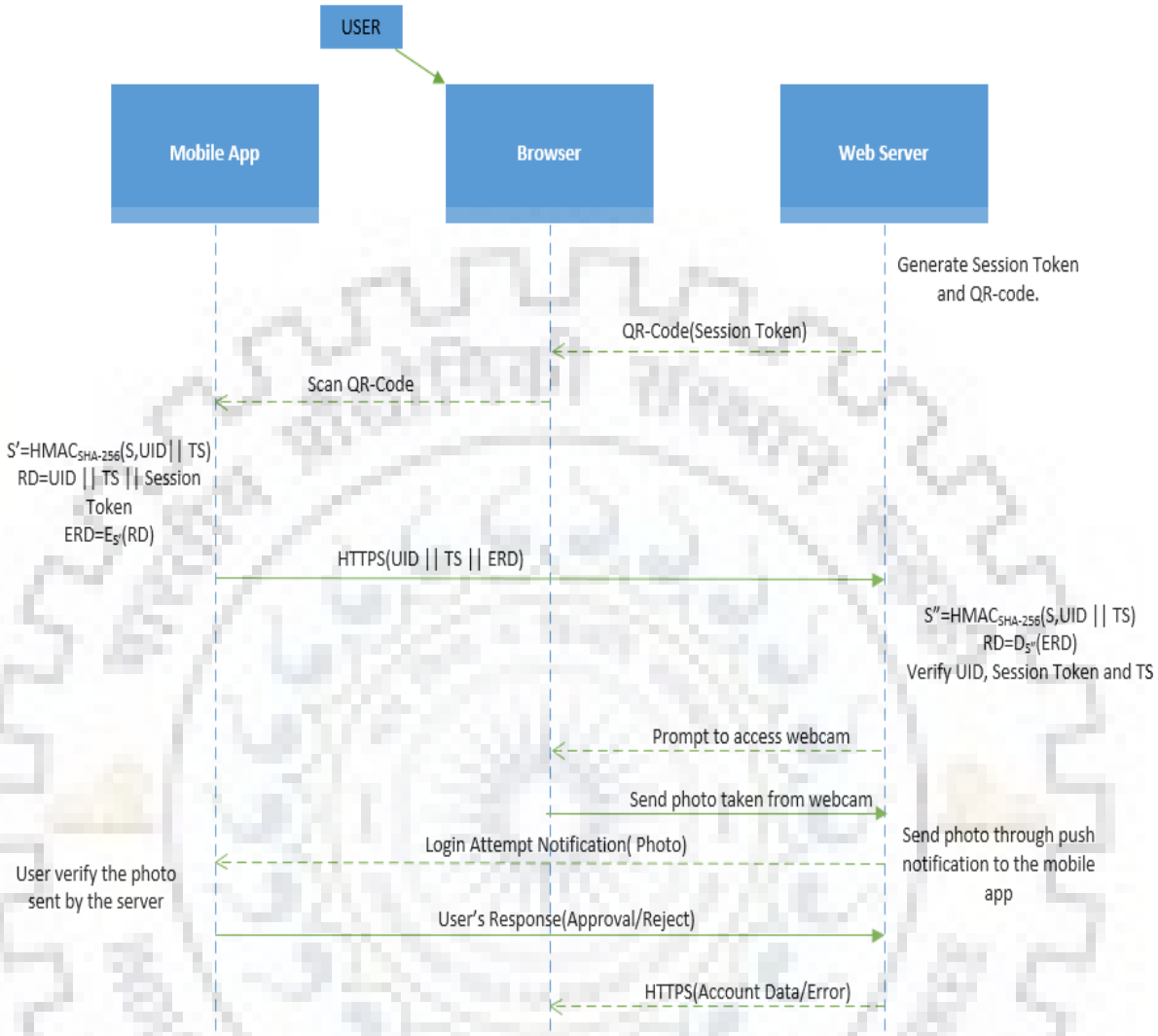


Fig.3.3 Login Phase

The login session is alive for a particular time period, after which the session will expire and the web server will redirect the user back to the home page displaying a new QR-code and thus preventing the attacker from using the image taken by the webcam in the later stage for authentication.

3.1.6 Recovery

The user can recover his/her account in case the mobile phone is lost or stolen. The user can request for recovery by using the registered email address. Once the server receives the request for recovery, a link will be shared with the user via the registered email address, where the user will be asked for the password. Once the password is verified, then the previous secret key which was stored in the Android Keystore API [35] will be deleted and a newly generated secret key will be sent to the user's new smartphone, which in turn will be stored in the Android Keystore API.

3.1.7 Storage Details

The proposed scheme stores certain information on the smartphone as well as web server per user. The details stored are as follows:

a) **Server side:** The proposed scheme requires the web server to store user's user-id (U_{ID}), email-id, phone number, hash password (h_{pwd}), secret key (S).
b) **Client side (smartphone app):** The proposed scheme stores the user-id (U_{ID}) and the secret key (S) in a secure storage which is encrypted by the Android Keystore API [35].

3.2 Implementation Details

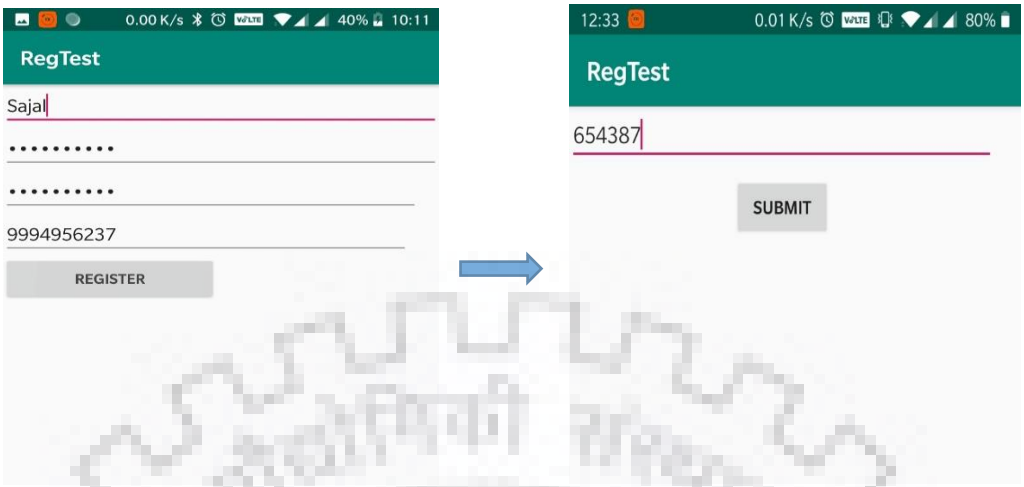
The proposed scheme consists of the following components: smartphone application, a web server that understands the proposed authentication protocol and a database for storing user's account details.

a) **Smartphone Application:** The smartphone application for the proposed scheme has been implemented using Android Studio and it is compatible with Android 4.4 and upward platforms. The smartphone application for the proposed protocol is responsible for user registration phase as well as the login phase. Built-in JAVA crypto and security libraries have been used for salt generation, AES encryption and decryption. For AES encryption, 256-bit key has been used. Zxing [37] version 3.3 has been used for generating the QR-code on the server side and for decoding the QR-code in the Android application. Android Keystore API [35] has been used for storing the shared secret key securely inside the user's smartphone.

b) **Server:** The web server for the proposed scheme has been implemented in JAVA using the RESTful web services with the help of the Spring framework. In the registration phase, the web server sends one time password (OTP) to the registered phone number using the Twilio SDK version 7.35 [38], which enables the developers to programmatically send and receive text messages using its web services. On the server side also, built-in JAVA crypto and security libraries have been used for generating the shared secret key (S), for generating hash password, for decrypting encrypted request data (ERD). In the login phase, the web server access the webcam of the client machine using the library Webcam Capture version 0.3.12 [39]. The captured photo is sent by the web server to the smartphone app using the Firebase API [36], which provides developers a platform for sending push notifications to the smartphone application. The mobile app has been registered with the Firebase API and therefore, whenever the user installs the mobile app for the proposed scheme, then a registration token is generated which uniquely identifies that particular smartphone. Using the registration token of the user's smartphone, the captured photo from the webcam is sent to that particular smartphone via push notification.

c) **Database:** The database used for storing the user's account information i.e., user-id, phone number, hashed password and secret key, is MySQL. JDBC (Java Database Connectivity) is a JAVA API which has been used in the implementation for connecting the database with the web server and for executing database queries.

Figure 3.4 (a) shows the registration page in the smartphone app. The user enters the userid, password and their phone number. The user is then directed to a page as shown in figure 3.4 (b) asking for OTP (one-time password) sent on the phone number mentioned in the form shown in figure 3.4 (a). After the user enters the OTP and submit the details, the user details are stored in the database as shown in figure 3.5.



(a) (b)

Fig. 3.4 Registration form: (a) User entering registration details, (b) User entering OTP.

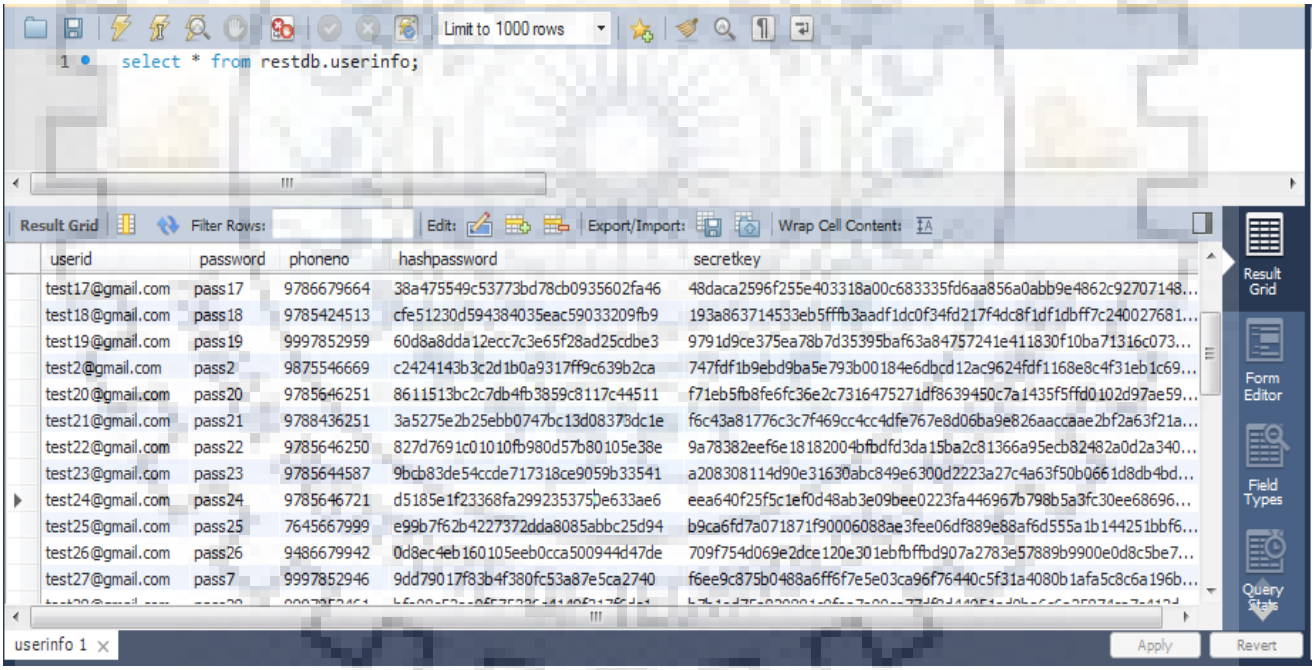


Fig. 3.5 Database storing users' account details

Figure 3.6, 3.7 shows the snapshots of smartphone application and website following the proposed scheme. Initially the user opens the website (Figure 3.6) and then scans the QR-code using the smartphone application (Figure 3.6). The user information is then sent from the mobile app to the

web server. The web server verifies the user information and clicks an image of the user using the client's desktop webcam. The webserver also adds a random text at a random position over the image taken (Figure 3.7). This image is then sent to the user's smartphone via push notification (Figure 3.7). The user then verifies the image and the embedded text and approves/rejects the login request depending on the authenticity of the image as shown in figure 3.7.

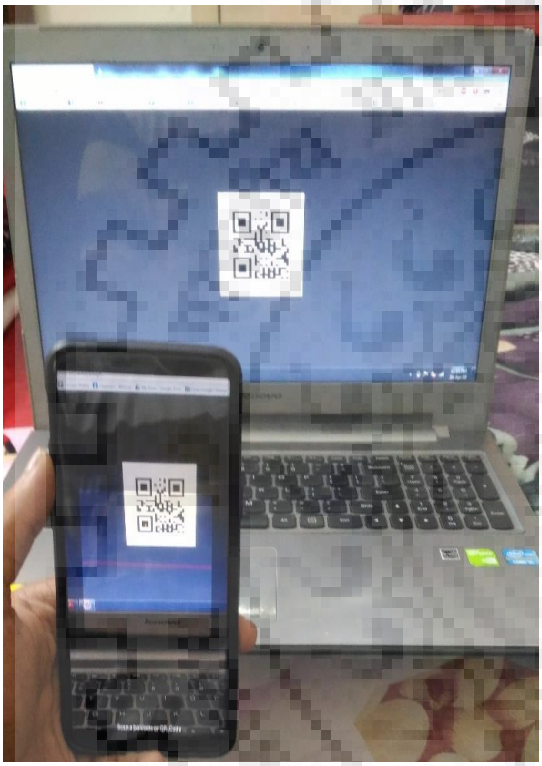


Fig.3.6 User scanning the QR-code using the smartphone app

Fig.3.7 Image taken by user's webcam sent to the smartphone app via push notification for user's approval.

CHAPTER 4

TESTING AND EVALUATION

4.1 Test Setup

The following hardware and software were used for the implementation and testing of the proposed scheme:

a) **Smartphone:** A OnePlus 5T smartphone with Qualcomm Snapdragon 835 MSM8998 Chipset, Octa-core CPU (Four 2.35GHz Kryo 280 Performance cores and four 1.90GHz Kryo 280 Efficiency cores), 6GB RAM and Android Pie operating system having OxygenOS version 9.0.4.

b) **Client's PC:** A desktop running Windows 7 Ultimate 64-bit operating system with Mozilla Firefox (Version 66.0.2), a webcam and an Intel® Core™ i5-3230M CPU @ 2.60GHz with 8GB of RAM.

c) **Server:** The website for testing the registration and login phase has been hosted on a desktop running Windows 10 64-bit operating system on an Intel® Core™ i7-3770 CPU @ 3.40 GHz with 8GB of RAM.

d) The website has been written in JAVA and hosted on Apache Tomcat 8.5.39 server.

e) **Database:** MySQL server version 8.0.13 has been used for storing the user's information on the server side.

4.2 Performance Evaluation

This section evaluates the performance of the proposed scheme in terms of the time taken by various operations in registration and login phase. The performance of the proposed scheme has also been evaluated in terms of the resources i.e., CPU and Memory, utilized by the proposed scheme. For all the experiments below, the client's machine is connected to a Wi-Fi router which

in turn is connected to 100 Mbps LAN, whereas the smartphone is connected either to the same Wi-Fi or a 4G network.

4.2.1 Timing Analysis

Time taken by the proposed scheme in the user registration phase as well as the login phase has been evaluated by performing the experiment 20 times and then taking the average of all the values.

The time required for the user registration ($T_{Registration}$) can be expressed using the following expression:

$$\mathbf{T_{Registration} = T_{OTP} + T_{MD5} + T_K + T_E + T_S + T_{DB}}$$

Where,

T_{OTP} = Time taken to send the OTP from the server and receive the OTP on the user's phone.

T_{MD5} = Time taken to generate hash password from the password entered by the user.

T_K = Time taken to generate 256-bit AES key using PBKDF2.

T_E = Time taken for 256-bit AES encryption

T_S = Time taken to send the registration details entered by the user on the smartphone app to the server.

T_{DB} = Time taken to store registration metadata securely in the app database using Android Keystore API.

The average value of T_R i.e. the registration phase for 20 trials is 13.885 seconds when the smartphone is to Wi-Fi and 14.704 seconds when the smartphone is on 4G network. Table 4.1 summarizes the time taken by each step in registration phase excluding the time taken by the user to fill the details.

Table 4.1 Average time taken by registration phase

	Time (sec)
T_{OTP}	12.21
T_{MD5}	0.025
T_K	1.285
T_E	0.048
T_S (Wi-Fi)	0.106
T_S (4G)	0.925
T_{DB}	0.211
Registration Time ($T_{Registration}$) (Wi-Fi)	**13.885**
Registration Time ($T_{Registration}$) (4G)	**14.704**

Similarly, the time required by the user for logging (T_{Login}) into the website using the proposed authentication protocol will be:

$$\mathbf{T_{Login} = T_{FQR} + T_{SQR} + T_E + T_D + T_{CAM} + T_{PN} + T_F}$$

Where,

T_{FQR} = Time taken to fetch the metadata from the server and display the QR-code on the webpage.

T_{SQR} = Time taken to scan the QR-code by the user's smartphone app.

T_E = Time taken for 256-bit AES encryption

T_D = Time taken for 256-bit AES decryption

T_{CAM} = Time taken to capture the image from the webcam

T_{PN} = Time taken for sending and verifying image via push notification

T_F = Time taken to fetch the user's account after submitting the login details.

The average value of T_L i.e. the login phase for 20 trials is 4.744 seconds when the smartphone is to Wi-Fi and 5.222 seconds when the smartphone is on 4G network. Table 4.2 summarizes the time taken by each step in login phase.

Table 4.2 Average time taken by login phase

	Time (sec)
T_{FQR}	0.405
T_{SQR}	0.019
T_E	0.674
T_D	0.528
T_{CAM}	2.309
T_{PN} (Wi-Fi)	0.249
T_{PN} (4G)	0.727
T_F	0.56
Login Time (T_{Login}) (Wi-Fi)	**4.744**
Login Time (T_{Login}) (4G)	**5.222**

4.2.2 Resource Usage

The resources utilized by the Android App of the proposed scheme is determined by using the profiling tool of Android Studio. Figures 4.1, 4.2, 4.3, 4.4 show that the recorded values of the minimum and maximum CPU utilization in the registration and login phase are 1.2%-19.8% and

1.4%-27.3% respectively. Figure 4.5 shows that the average memory used by the Android app for the proposed scheme is 46MB whereas the maximum memory used by the smartphone app is 149MB. From the statistics of the time recorded and resource utilization, it is evident that the proposed scheme shows favorable performance and therefore, the proposed scheme can be utilized for logging into websites.

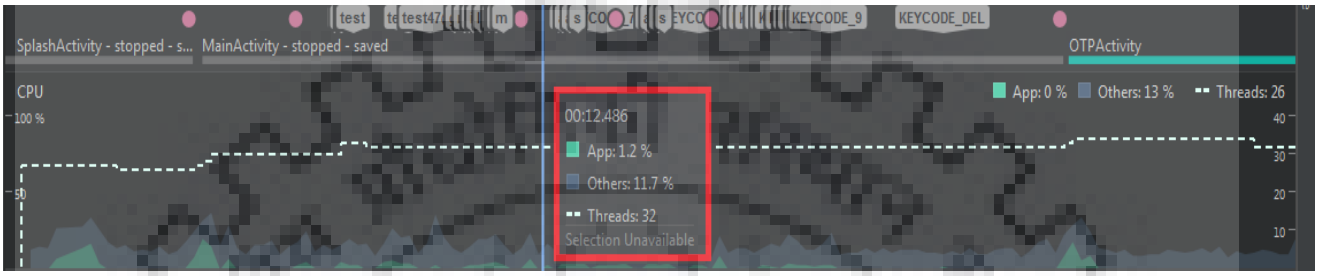


Fig.4.1 Minimum CPU Utilization (Registration)

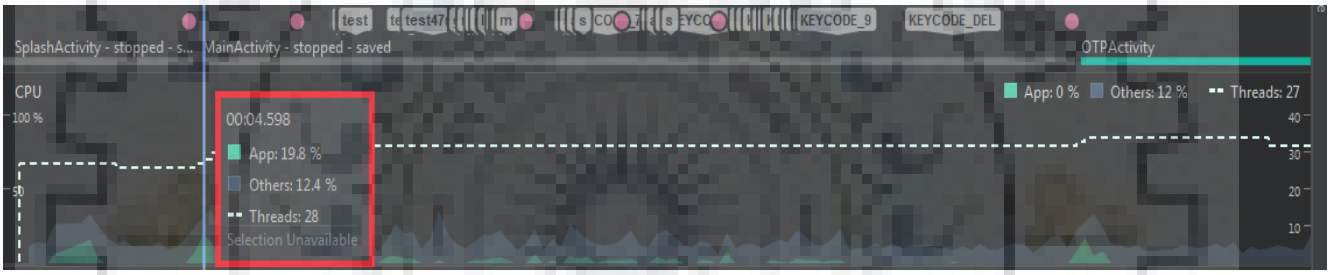


Fig.4.2 Maximum CPU Utilization (Registration)

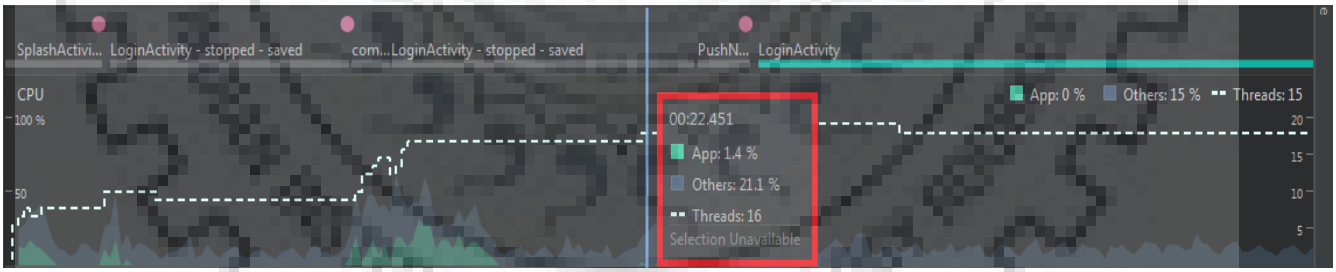


Fig.4.3 Minimum CPU Utilization (Login)

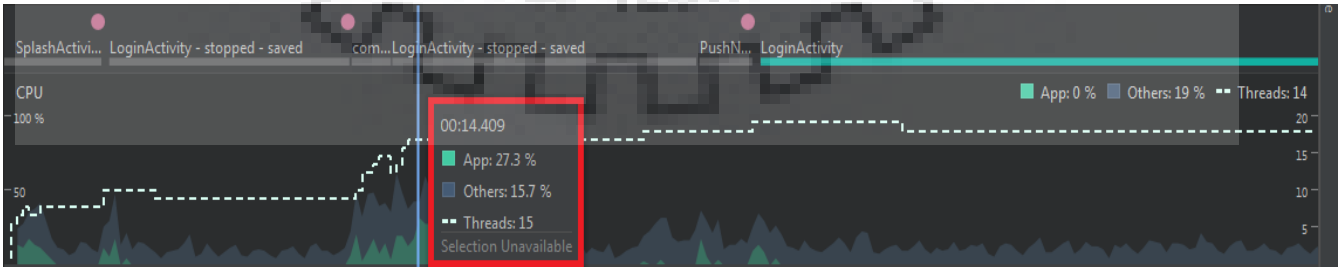


Fig.4.4 Maximum CPU Utilization (Login)

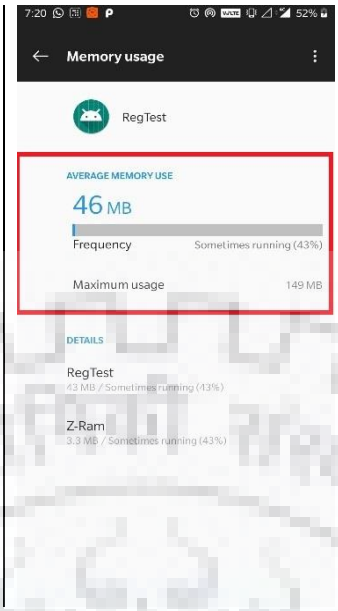


Fig. 4.5 Memory usage of the smartphone app

4.2.3 Security Analysis

a) **Traditional Phishing Attacks:**

Attack: The attacker can lure the user by directing him/her to the phishing website and then obtain personal information by storing them in the database for later use.

Mitigation: The proposed scheme is secure against the traditional phishing attacks as this method of login does not require the user to enter username and password. The user information is stored in the mobile app and then user information will be sent to the web server using the mobile app only. Thus, traditional phishing attacks requiring user information over the phishing website will not be able to obtain any credentials from the user on the phishing website.

b) **RT MITM Phishing Attack:**

Attack: In RT MITM phishing attack, the attacker relays the information from the authentic website to the user and vice versa in real time.

Mitigation: The proposed scheme is secure against the RT MITM phishing attack. This can be explained by taking the following attack scenario. The attacker relays the QR-code from the authentic website to the phishing website. The user scans the QR-code from the phishing

website and the mobile app will send the information to the server. The browser on the attacker's screen will request for the picture to be taken from the webcam. Suppose the attacker shows a similar prompt to the user and the user takes a photo from the webcam. Now the picture received by the attacker cannot be sent to the web server as the picture has to be taken at that time from the webcam. Assuming there are no desktop malwares or bots coordinating in real-time, the attacker will not be able to get the user's image and send it to the server and thus the proposed scheme is safe from RT MITM phishing attack.

c) **CR MITM Phishing Attack:**

Attack: In CR MITM phishing attack, the attacker relays his/her desktop over the user's terminal using applications such as TeamViewer.

Mitigation: The proposed scheme will be secure against the CR MITM phishing attack. The reason is that after the user will scan the QR-code relayed on his/her desktop by the attacker, the second factor of the authentication scheme requires the user to take a picture from the webcam and thus, the webcam will be open on the attacker's machine and not the user's machine.

d) **Malicious Extension Based Phishing Attacks:**

Attack: The malicious browser extensions ask for permission from the user to provide some functionality to the users in the foreground. Now, the same set of permissions are used to carry out an attack in the background so as to obtain user's credentials. The malicious browser extensions can perform keylogging, screen logging or password sniffing in the background.

Mitigation: The proposed scheme will also be able to prevent malicious extension based attacks such as keylogging, password sniffing because these attacks sends the webpage information to the attacker and thus they record the credentials entered or record the mouse-click coordinates, but in the proposed scheme, the user will never be asked to enter his username and password during the web login phase. Therefore, this attack will also be prevented.

e) **App Spoofing:**

Attack: The attacker can install this spoofed app on the user's smartphone and lure the user into entering his/her credentials over this app.

Mitigation: The proposed scheme is secure against app spoofing. If the attacker installs spoofed app on the user's mobile and when the user will use the spoofed app for web login, then the mobile app will not have the secret key (S) in the Android Keystore and thus, will not be able to complete the authentication process.

f) **Guessing Attack:**

Attack: An attacker can guess the login details by repeating the login attempts.

Mitigation: The proposed scheme is secure against this attack as the scheme does not require the user to enter any credentials so there will not be any credentials to be guessed by the attacker. If the attacker still tries to login, then the attacker must guess the secret key stored in the Android Keystore API [35] correctly. The secret key is 256-bit and thus it requires 2^{256} attempts on token generation which is practically not possible. The number of unsuccessful attempts by the attacker can be fixed to a certain number after which an email will be sent to the user regarding the unsuccessful login attempts.

g) **Internal Observation:**

Attack: The attacker can obtain user's details by intercepting user's input with the help of keylogging. The attacker can intercept communication between the user and the verifier.

Mitigation: Keylogging tools cannot get any user login details as it is not inputted by the user. The attacker cannot access the mobile database which contains the secret key (S) as it is encrypted using Android Keystore API. Due to this reason, the attacker cannot gain any information even after intercepting the message between the user and the server as they are encrypted by the stored secret key (S).

h) **Leaks from external verifiers:**

Attack: The attacker can manipulate the third party verifiers and use the provided information for login.

Mitigation: Since the proposed scheme does not have any third party verifier, therefore it is not vulnerable to this attack.

i) **Physical Observation:**

Attack: The attacker can observe the user multiple times at the time of authentication. Shoulder surfing is an example of physical observation.

Mitigation: The proposed scheme is secure against this attack as the user does not enter any login details while authenticating into the website. The attacker won't get any login details by physically observing the user. Moreover, the secret key is stored in the Android Keystore API which is not visible and thus, the attacker won't be able to login into the website by physical observing the user.

j) **Targeted Impersonation:**

Attack: An attacker can break the user's account by using the personal information of the user such as date of birth, security question, etc.

Mitigation: The proposed scheme is secure against this attack as the scheme does not require the user to enter any details and use the secret key encrypted by the Android Keystore API as the first factor for authentication which cannot be accessed by the attacker who is impersonating the user by leveraging his/her personal information.

k) **Theft of Mobile Device:**

Attack: The attacker can steal user's mobile device and attempt to login into the website.

Mitigation: The mobile device is assumed to be protected by password, pin or pattern lock and thus the attacker won't be able to access the app. Moreover, the victim will report about the stolen device via email and a new secret key will be generated and shared with the user, thus invalidating the old secret key stored in the stolen phone and stopping the attacker to access the user's account.

l) **Session Hijacking:**

Attack: The attacker can steal session information by eavesdropping on the connection between the user and the server. The attackers can inject this session information into their browser to access the users' accounts.

Mitigation: The proposed scheme requires the website to use HTTPS connection for all communication. This prevents the attackers from accessing the session data from the encrypted stream. But after authentication, the management of authentication cookies is left to the websites. The protocol for secure cookie management can be implemented by the websites for this purpose.

4.2.4 User Survey

A user survey was conducted in order to understand the users' opinion about different authentication schemes that can be used for login over websites on desktop. A total of 20 people from different age group having different computer proficiency participated in the survey. A detailed introduction was given for the following 7 authentication schemes: (1) U-PWD, (2) U-PWD and OTP, (3) QR-code based, (4) Graphical PWD based, (5) Hardware token based, (6) Push notification based login scheme, and (7) Proposed Scheme. Figure 4.6 shows the age range and computer proficiency percentages of the participants respectively.

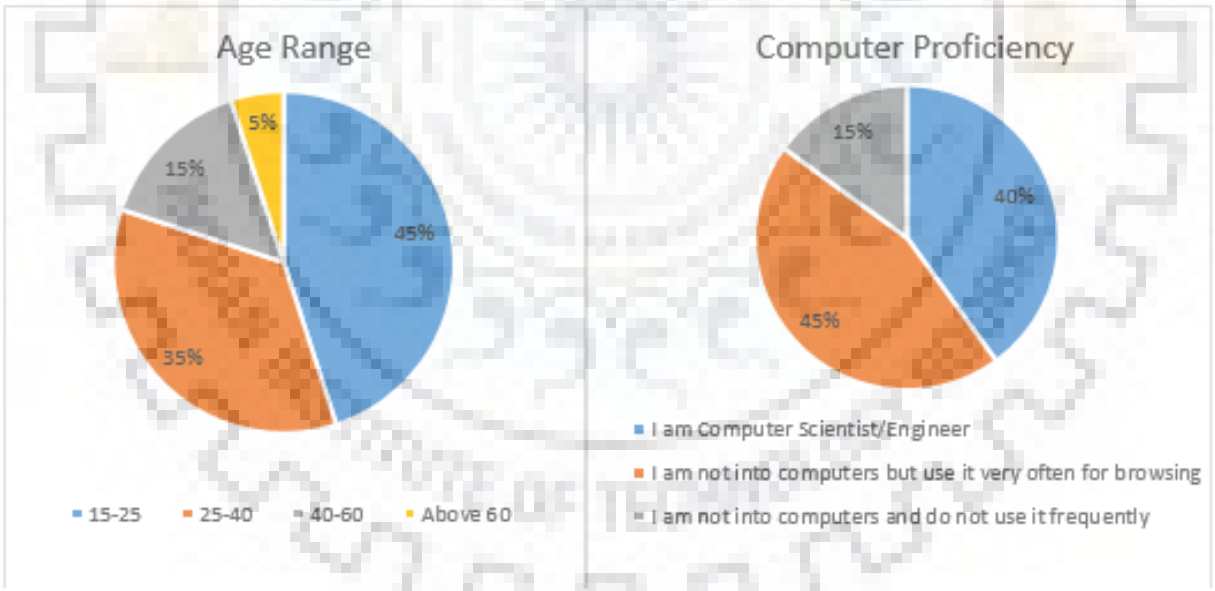


Fig.4.6 Participants' information representing the age groups and their computer proficiency

It should be noted that only few people having age above 40 years participated in the survey and thus the survey does not reflect the opinion of a majority of older people. Most of the people that

participated in the survey are probable users of the scheme and thus opinion of the people who do not use computer frequently will be done in the future. All the participants were given a brief introduction about all the 7 schemes and then the following questions were asked, out of which the users were given a choice to select more than one authentication scheme in questions (b), (c) and (d):

a) Rate individual schemes from 1 to 5 (1- Very difficult, 5- Very easy) with respect to "Easy to Learn and Use" in your daily life.
b) Mention the schemes that the user considers to be secure against known attacks.
c) Mention the schemes that will be preferred by the users to login over a website containing data of high importance such as banks, etc.
d) Mention the schemes that the user thinks maintain a balance between “Easy to learn & use” and “Security” and should be standardized by companies.

Figure 4.7 shows the result for question 1, in which each authentication scheme has an overall score on a scale of 100. From the figure 4.7 shown, it can be concluded that the users find U-PWD scheme easiest to learn and use. The proposed scheme comes at third position after OTP based schemes with a score of 74 out of 100. It can also be observed that the hardware token based schemes has the least score as the users have to carry an additional token for the purpose of authentication.

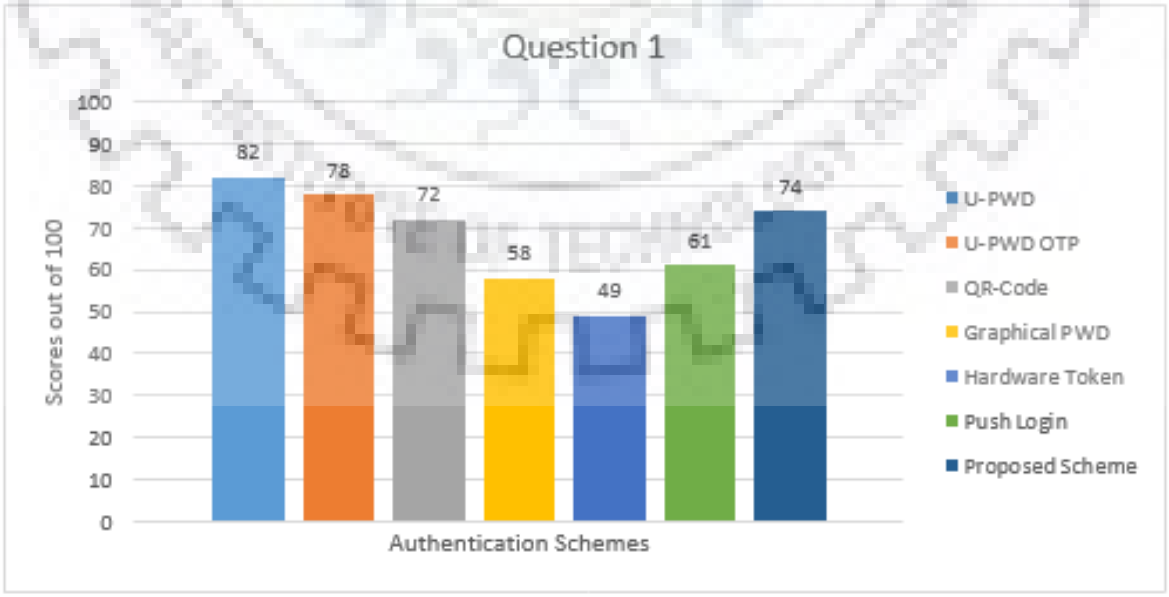


Fig.4.7 User rating for each scheme with respect to ease of use

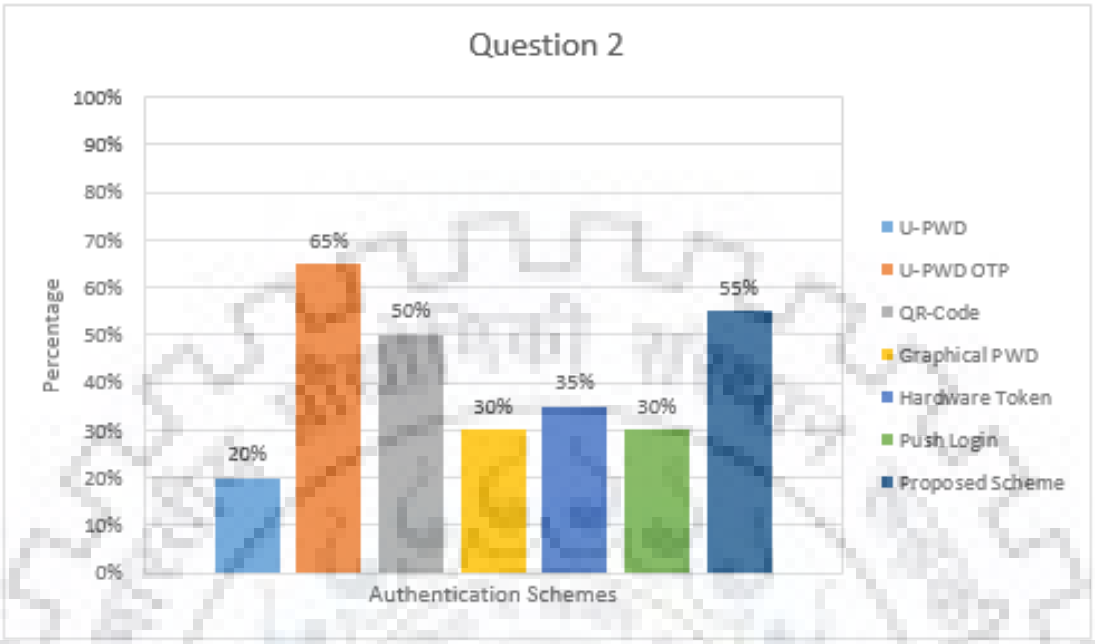


Fig.4.8 Data representing users' confidence on security of respective authentication schemes

For the 2nd question, the proposed scheme scored 55% which is more than most of the schemes taken for comparison. The proposed scheme was voted second out of all the schemes in 2nd question which shows the users' confidence in the security of the proposed protocol. The graph in figure 4.8 also shows that the users were confident the most about OTP based authentication scheme whereas they were least confident about the U-PWD authentication scheme.

The graph in figure 4.9 shows the result for the 3rd question, in which the users were asked to choose the schemes that should be used to secure the data in banks or financial accounts. The graph shows that according to the users, the proposed authentication protocol can be used in banks to secure the data. The users preferred the proposed authentication protocol more than other schemes. OTP based authentication schemes was voted the most as it was chosen by 65% of the total participants. The proposed scheme, on the other hand was chosen by 50% of the participants. The graphical password based scheme and the push login based scheme received almost equivalent votes and the basic U-PWD scheme received the least votes as it is not much secure.

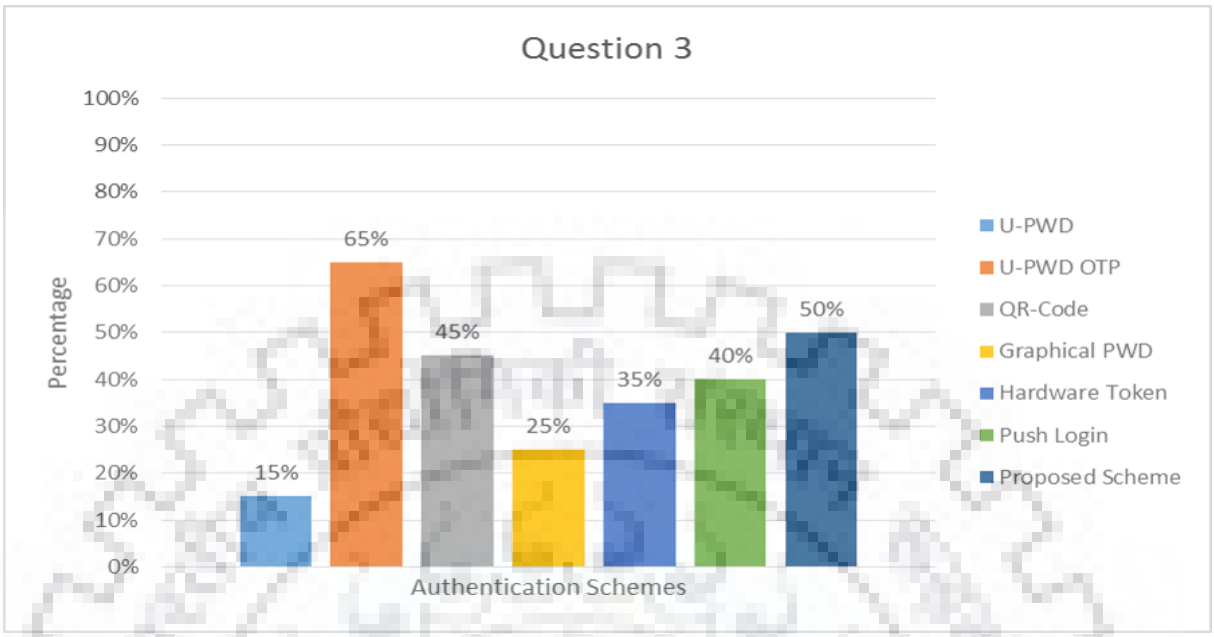


Fig.4.9 Data representing users' preference of authentication schemes to secure data

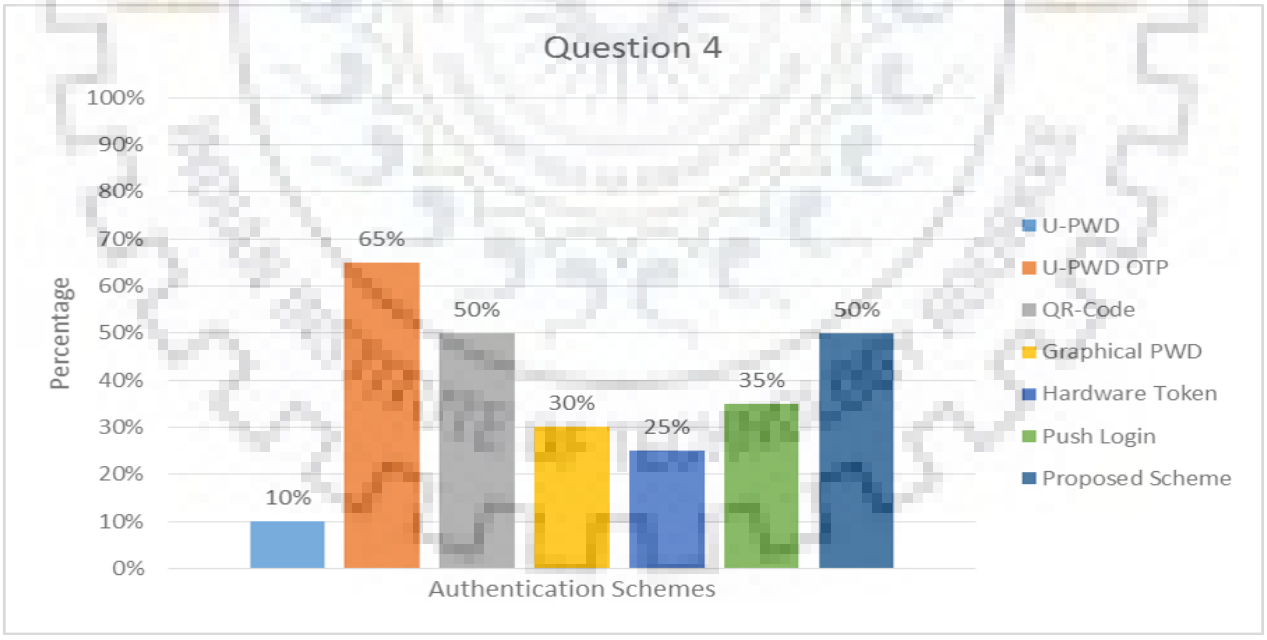


Fig.4.10 Data representing users' opinion on authentication schemes which create a balance between ease of use and security.

The result for the 4th question can be seen in the figure 4.10. The graph clearly shows that the 65% of the users thought that U-PWD with OTP offers a great balance between usability and security. It can also be seen that the proposed authentication scheme and QR-code based scheme received equal amount of votes i.e. 50% and after the OTP based scheme, the proposed scheme is considered to have balance between usability and security and thus the proposed scheme can be used for login over different websites.

After the survey, we also interacted with the participants and concluded that since the participants have not used the proposed scheme, they were not sure about selecting it in the first question. Also, most of the participants were unaware of the attacks which can compromise the authentication schemes. The survey respondents were unaware of the phishing attacks that can be easily launched on the OTP based authentication scheme. They had the perception that OTP provides security against all the attacks. From the above user survey, it can be concluded that the proposed scheme provides usability and security as compared to hardware token based schemes, push login, QR code based schemes, etc. Further details of the survey can be accessed at https://forms.gle/gz8QrR2JRDhEq3CM7.

4.3 Comparison

This section compares the proposed scheme with the existing authentication schemes on the basis of usability, security and using Bonneau et al.'s framework [28].

4.3.1 Usability

The proposed scheme has been compared with the existing schemes based on the usability. The usability of an authentication scheme considers the following parameters for evaluation- the number of tokens required by the authentication scheme and the number of authentication tokens that the user has to remember. This comparison also takes any additional software or hardware needs into consideration. Smartphone is required by most of the existing schemes as it is carried by all the internet users. Some of the existing schemes require particular module or driver to be installed on the client machine, hardware token or trusted third party, etc. The proposed scheme requires PC camera which is easily available in all the desktops, especially laptops. Also, the

proposed scheme does not require the user to remember any authentication token and thus, it is more user-friendly as compared to other authentication schemes. The comparison based on usability also considers the need for internet on the phone as one of the factors for measuring usability in different authentication schemes to determine the cost incurred while using the scheme. The proposed scheme requires the internet on the phone, which has become synonymous with smartphones. Table 4.3 describes the comparative analysis based on usability.

Table 4.3 Comparison in terms of usability

SNO.	Scheme	Tokens used by the scheme	Tokens to be remembered by the users	Additional needs	The need for Internet on the phone
1.	Google 2 Step [15]	3- U,PWD,OTP on SP	2- U, PWD	Cellphone	N
2.	SAASPASS [14]	3- U,PWD, OTP on App	2- U, PWD	Smartphone	Y
3.	Xie et al. [16]	4- U, PWD, DH Public (g, p), Private Up	2- U, PWD	PC Cam, Smartphone	N
4.	Kim et al. [17]	4- U, PWD, Session ID, Secret Key	2- U, PWD	Smartphone with GPS	Y
5.	Mukhopadhyay et al. [18]	3- U, PWD, Secret Key in SP	2- U, PWD	Smartphone, TTP	Y
6.	Dodson et al. [19]	4- U, PWD, Secret Key, QR Code	0 – NIL (QR Code Scan)	Smartphone	Y
7.	Leung et al. [20]	4- U, PWD, Secret Key, OTP CAPTCHA	2- U, PWD	NIL	NA
8.	Zhu et al. [21]	3- U, SALT, PWD, CAPTCHA	2- U, PWD	NIL	NA
9.	Tricipher [25]	4- U, PWD, TPM Secret Key, TACS credential	2- U, PWD	CAPI Driver, Separate Hardware, TPM	N
10	RSA SecurID HW Token [27]	4- U, PWD, HW token information, PIN	2- U, PWD	Separate Hardware	NA
11.	Yubikey U2F [26]	5- K_{PUB}, K_{PRIV}, Counter, U, PWD	2- U, PWD	Separate Hardware	NA
12.	Push Login [22, 49]	3- U, PWD, SP	1 – U	Smartphone	Y

13.	Password Managers [23, 24]	3- U, PWD, the master key of the password manager	Master PWD	NIL	NA
14.	U-PWD [28, 44]	2- U, PWD	2- U, PWD	NIL	NA
15.	Ritwik et al. [46]	3- U,PWD, Token from server	2- U, PWD	Smartphone	Y
16.	Proposed Scheme	3- U, Secret key stored on SP, photo taken by PC Cam	0 – NIL (QR Code Scan)	Smartphone, PC Cam	Y

Note: *TPM: Trusted Platform Module, U: Username, SP: Smartphone, K_{PUB}, K_{PRIV}: (Public and Private Key pairs), NA: Not Applicable, DH: Diffie-Hellman, TACS: Tricipher Armored Credential System.*

4.3.2 Security

This section compares the proposed scheme with the existing schemes based on the security they provide against the attacks discussed in section 3.1.2.

a) RT MITM and CR MITM phishing attacks

U-PWD [28, 40] and OTP/PIN based authentication schemes such as Google 2-step [15] and SAASPASS [14] are vulnerable to RT MITM as well as CR MITM phishing attacks as the attacker can easily obtain the user credentials with the help of the phishing website. The attacker can deceive the user by showing him/her an exact copy of the authentic website and relay the credentials entered by the user on the phishing website in real time to the authentic website. A detailed simulation of RT MITM attack on the Google 2-step authentication scheme has been explained in section 2.3. The attacker can also launch CR MITM phishing attack on the OTP/PIN based scheme, simply by relaying his/her remote desktop over client's terminal and the user is lured to enter the credentials, which is actually entered on the attacker's remote desktop.

QR-code based schemes also fail to stand against RT MITM and CR MITM attacks. The scheme proposed by Xie et al. [16] can be attacked using spoofed malicious browser extension. The attacker can install a spoofed malicious browser extension on the user's PC. In this way, the attacker will obtain the credentials, which the attacker will relay to the CamAuth extension. The CamAuth extension will send the user information to the server and will also initiate Diffie-

Hellman exchange. The server verifies the credentials and sends a barcode to the attacker's browser, which the attacker relays to the malicious extension. The user scans the barcode using his/her smartphone and then a barcode, containing the vouch request, is displayed on the user's smartphone which is scanned by the PC cam. This barcode is received by the attacker using the malicious extension. The attacker transfers this barcode to his/her smartphone and shows this barcode to the PC cam. In this way, the server receives the barcode from the attacker's smartphone and thus, user's account will be sent by the server to the attacker's browser. However, Xie et al.'s scheme [16] is secure against CR MITM attack because the user's PC cam cannot be accessed by the attacker. Similarly, the QR-code based scheme proposed by Kim et al. [17] is also not secure against RT MITM and CR MITM attack. The reason is that the IP address present in the QR-code can be spoofed and a login request can be sent by the attacker to the server. The server, on receiving the request, will send the QR-code containing IP address and session id, to the attacker's browser. The attacker, then relays this QR-code to the phishing website which is open on the user's browser. The user scans the QR-code and the IP address is verified by the QRA app. Thus, the server receives the verification token from the QRA app and sends the account to the attacker's browser. The scheme proposed by Mukhopadhyay et al. [18] is also vulnerable to these attacks because the credentials entered by the user can be easily obtained by the attacker using the phishing website. The QR-code, generated by the server contains IDP (Identity Provider) generated challenge encrypted by the user's secret key and it is sent to the attacker's browser. The attacker relays the QR-code to the phishing website and it is scanned by the user's smartphone. The smartphone will decrypt the challenge using the secret key and send the response to the server. Thus, the login process gets completed and the attacker gets access to the user's account. In the same way, Dodson et al.'s scheme [19] is also vulnerable to RT MITM as well as CR MITM attack. The attacker relays the QR-code to the phishing website, which is scanned by the user's smartphone. After scanning, the smartphone verifies the browser session and user information is then sent to the server, thus completing the login procedure. A detailed simulation of RT MITM attack on QR-code based scheme has been explained in section 2.3.

Hardware token based schemes such as Tricipher [25] and Yubikey using U2F [26] are secure against RT MITM and CR MITM phishing attacks due to the use of multipart credentials.

However, RSA SecurID soft token/hardware token [27] are not safe from these attacks as the attacker can obtain the RSA passcode via the phishing website.

Graphical password based authentication scheme proposed by Leung et al. [20] is secure against RT MITM attack because relaying the user's mouse click coordinates on moving CAPTCHA is quite difficult and there might be a difference in screen resolution of the attacker's desktop and user's desktop. However, this scheme is not safe from CR MITM attack. Zhu et al.'s scheme [21] is not safe from RT MITM as well as CR MITM attack because the user's click coordinates can be recorded and the attacker then maps these coordinates to his/her screen and click the corresponding points on the CaRP displayed on the authentic webpage.

Push notification login based schemes [22, 45] are vulnerable to RT MITM and CR MITM attack because the attacker can relay the credentials entered by the user and the user will then approve the push notification received on the smartphone app.

Password managers [23, 24] are secure against RT MITM and CR MITM attacks as the user's credentials are automatically sent by the password managers and the user is not required to enter them. However, if wrong credentials are entered by the attacker, then it will prompt the user for re-entering the username and password, thus leading to RT MITM and CR MITM attack.

The proposed scheme safeguards the user from RT MITM phishing attack because even after relaying the QR-code, the attacker cannot send the user's photo taken from the user's webcam to the server. The reason is that the image has to be taken by the attacker's webcam. Moreover, after the user scans the QR-code from the smartphone app, the attacker's webcam will immediately click the attacker's photo using his/her webcam and send that photo to the user via push notification, thus alerting the user. The proposed scheme is also safe from CR MITM phishing attack as the attacker cannot access the user's webcam.

b) Malicious browser extension based phishing attacks

The malicious browser extension based phishing attack includes key logging, screen logging and password sniffing. The attacker can break Google 2-step [15], U-PWD [28], SAASPASS [14], RSA SecurID software/hardware [27] token using key logging and password sniffing because in the above mentioned schemes, all the credentials are entered on the website including the second

factor of authentication which can be fetched via the malicious browser extension. The attacker can also break the password managers by installing a malicious browser extension on the client's terminal which can sniff the password before it is sent to the server.

Xie et al.'s scheme [16] is safe from key logging and password sniffing because the malicious browser extension cannot access the information entered by the user on the CamAuth extension due to the same origin policy. However, the attacker can access the QR-code using the screen logging but the complete authentication will not be compromised. Kim et al. [17] and Dodson et al. [19] safeguard the users from these attacks as it does not requires the user to enter any credential.

Leung et al.'s CAPTCHA based scheme [20] is safe from key logging and password sniffing because flash-based moving OTP CAPTCHA is used. On the other hand, the attacker can break Zhu et al.'s scheme [21] and obtain the user input via screen logging.

Since Zhu et al. [21], Tricipher [25], Mukhopadhyay et al. [18], Ritwik et al. [46] and push login based schemes [22, 45] either use trusted device or second authentication factor, therefore the attacker can only obtain the user's authentication token or identification but cannot compromise these schemes.

The proposed scheme is safe from these attacks because the user does not need to enter any credentials and thus the malicious browser extensions won't be able to sniff any information.

c) App spoofing:

The spoofed smartphone application or extension can be installed by the attacker on user's smartphone or desktop, so as to obtain user's information. Ritwik et al. [46], Xie et al. [16], Google 2-step [15], SAASPASS [14], U-PWD, Kim et al. [17] and push login based schemes [22, 45] are vulnerable to this attack as the spoofed app can be installed on the user's device and the credentials entered by the user are stored in the spoofed app, which are received by the attacker. Zhu et al. scheme [21] is secure against this attack as it does not use any app or extension. Mukhopadhyay et al. [18], Tricipher [25] and Dodson et al. [19] are safe from this attack as these scheme have at least one part of the authentication token stored in the user's device and not entered in the app. The proposed scheme is also secure against app spoofing because the secret key (S) is stored in the app encrypted by Android Keystore API [35], which will not be available in the spoofed app. The

comparison of the proposed scheme with other existing schemes based on security has been summarized in Table 4.4.

Table 4.4 Comparison in terms of security

SNo.	Scheme	RT MITM	CR MITM	Key logging	Screen logging	Password sniffing	App spoofing	Secure Count
1	Google 2 step [15]	×	×	×	●	×	×	0
2	SAASPASS [14]	×	×	×	●	×	×	0
3	Xie et al. [16]	×	✓	✓	●	✓	×	3
4	Kim et al. [17]	×	×	✓	✓	✓	×	3
5	Mukhopadhyay et al. [18]	×	×	●	●	●	✓	1
6	Dodson et al. [19]	×	×	✓	✓	✓	✓	4
7	Leung et al. [20]	✓	×	✓	●	✓	●	3
8	Zhu et al. [21]	×	×	●	×	●	✓	1
9	Tricipher [25]	✓	✓	●	●	●	✓	3
10	RSA SecurID HW token [27]	×	×	×	●	×	×	0
11	Yubikey U2F [26]	✓	✓	✓	●	●	✓	4
12	Push login [22, 45]	×	×	●	●	●	×	0
13	Password Managers [23, 24]	✓	✓	×	●	×	✓	3
14	U-PWD [28, 40]	×	×	×	●	×	×	0
15	Ritwik et al. [46]	✓	✓	●	✓	●	×	3
16	Proposed Scheme	✓	✓	✓	✓	✓	✓	6

Note: *✓: Secure, ×: Insecure, ●: Partial user information can be accessed but the attacker cannot compromise the scheme. Secure Count: Count of the ticks (✓) and thus, a measure of the security against attacks.*

4.3.3 Comparison using Bonneau et al. Framework [28]

Bonneau et al. [28] proposed a usability-deployability-security evaluation framework using which an authentication scheme can be evaluated. The framework evaluates the user authentication protocol using 25 parameters comprising of 8 usability, 6 deployability and 11 security parameters.

This section compares the proposed scheme with the existing authentication schemes using the Bonneau et al. framework [28]. This section also discusses various parameters of Bonneau et al. framework and which schemes offers the benefit for that particular framework parameter. Table 4.5 summarizes the comparative evaluation of the proposed scheme with the existing authentication schemes. Some of the values in the table have been taken directly from the study [28, 41] and some of the values have been updated considering the newly identified attacks.

Table 4.5 Comparison using Bonneau et al's framework [28]

			Google 2-Step [15]	SAASPASS [14]	Xie et al. [16]	Kim et al. [17]	Mukhopadhyay et al. [18]	Dodson et al. [19]	Leung et al. [20]	Zhu et al. [21]	Tricipher et al. [25]	RSA SecurID token [27]	Yubikey U2F [26]	Push Login [22]	Password Managers [23, 24]	U-PWD [28, 44]	Ritwik et al. [46]	Proposed Scheme
Usability	1	Memory wise effortless	×	×	×	●	×	✓	×	×	×	×	×	●	●	×	●	✓
	2	Scalability for users	×	×	×	×	×	✓	●	●	×	×	×	✓	✓	×	●	●
	3	Nothing to carry	●	●	●	●	●	●	✓	✓	×	×	×	●	✓	✓	●	●
	4	Physically effortless	×	×	×	×	×	✓	×	×	×	×	×	●	✓	●	✓	✓
	5	Easy to learn	✓	✓	●	✓	✓	✓	●	●	✓	✓	✓	✓	✓	✓	✓	✓
	6	Efficient to use	●	●	×	●	●	✓	×	×	✓	✓	✓	✓	✓	✓	✓	✓
	7	Infrequent errors	●	✓	×	×	●	●	×	●	✓	✓	✓	✓	✓	✓	✓	✓
	8	Easy recovery from loss	●	●	●	●	●	●	✓	✓	●	●	●	●	●	●	●	●
Deployability	9	Accessible	●	●	×	×	×	×	×	×	✓	×	✓	●	✓	✓	●	×
	10	Negligible cost/user	×	●	●	●	●	●	×	✓	✓	×	×	●	✓	✓	●	✓
	11	Server compatible	×	×	×	×	×	×	×	×	×	×	×	×	✓	✓	×	×
	12	Browser compatible	✓	✓	●	●	●	●	●	●	✓	✓	●	✓	✓	✓	×	✓
	13	Mature	✓	✓	×	×	●	●	×	×	●	✓	✓	✓	✓	✓	●	●
	14	Non-Proprietary	×	●	✓	✓	✓	✓	✓	✓	×	×	×	●	✓	✓	✓	✓
Security	15	Resilient to physical observation	●	●	✓	✓	✓	✓	●	●	●	✓	✓	●	●	×	✓	✓
	16	Resilient to target impersonation	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	×	×	✓	✓
	17	Resilient to throttled guessing	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	×	×	✓	✓
	18	Resilient to unthrottled guessing	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	×	×	✓	✓
	19	Resilient to internal observation	×	×	×	●	×	×	●	●	×	×	●	●	×	×	✓	✓
	20	Resilient to leak from other verifiers	✓	●	✓	✓	●	✓	✓	✓	●	●	●	✓	●	×	✓	✓
	21	Resilient to Phishing	×	×	×	×	×	×	×	×	✓	×	✓	×	✓	×	✓	✓
	22	Resilient to Theft	✓	✓	✓	✓	✓	×	✓	✓	✓	✓	✓	✓	×	●	✓	●
	23	No Trusted Third Party	✓	×	✓	✓	×	✓	✓	✓	×	×	×	✓	×	✓	✓	✓
	24	Requiring explicit consent	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	×	✓	✓	✓
	25	Unlinkable	✓	✓	✓	●	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		Benefit Offered Count	**11**	**10**	**10**	**10**	**9**	**14**	**11**	**12**	**13**	**12**	**13**	**14**	**14**	**13**	**16**	**18**

1. **Memory wise effortless:** A scheme offers this benefit, if the user doesn't have to remember any information while logging into the website. The scheme is granted Quasi-Memorywise-Effortless if the user has to remember one secret for everything. This benefit is not offered by most of the authentication schemes and therefore they are represented by ×, as they require the user to remember their username and password. Schemes such as Yahoo push login [22, 45], password managers [23, 24] require user to remember less credentials and the number of tokens to be remembered are reduced and thus, they are granted Quasi-Memorywise-Effortless. The proposed scheme and Dodson et al. [19] does not require the user to memorize any token for login and thus they completely offer this benefit.

2. **Scalable for users:** An authentication scheme offers this benefit, if the user is not burdened by logging into multiple accounts using the same scheme. OTP/PIN based schemes [14, 15] are not scalable for users because the user has to receive and enter the OTP every time while logging into the website which makes the login procedure cumbersome. Schemes proposed by Xie et al. [16], Mukhopadhyay et al. [18] and Kim et al. [17] are also not scalable because the user has to enter the username and password and also scan the QR-code during every phase of login. Hardware token based schemes are also not scalable for the users because they need a new token and password/PIN per each verifier. Dodson et al. scheme [19] and password managers offers this benefit as the user is not burdened with the login procedure. Leung et al. [20], Zhu et al. [21] partially offers this benefit as the user has to just click the password over the CAPTCHA. The proposed scheme also partially offers this benefit because the user does not have to remember any token or credential. The user has to scan the QR-code and verify the image sent via push notification.

3. **Nothing and quasi-nothing to carry:** Nowadays, since mostly all the users carry a smartphone, therefore carrying a smartphone is considered as quasi-nothing to carry. Since almost all the users have desktops with built-in webcams, therefore the proposed scheme partially offers this benefit. Hardware token based schemes does not offer this benefit because the user has to carry an extra hardware or device.

4. **Physically effortless:** This benefit is not offered by most of the schemes because they require the user to enter their username, password and then some other physical interaction. The push login based scheme [22, 45] and U-PWD [28] partially offers this benefit because the only physical interaction these schemes require from the user is to enter their username and password. The proposed scheme, Dodson et al. [19] and password managers [23] completely offers this benefit because the user does not have to enter any username and password. Moreover, the proposed scheme requires to just click once to verify the image sent via push notification, thus making it physically effortless.

5. **Easy to learn:** Most of the schemes that are taken for comparison with the existing schemes are easy to learn because an average user with basic computer knowledge can understand and recall how to use it. CAPTCHA based schemes partially offers this benefit because the user has to click the graphical passwords which makes the login process cumbersome. The proposed scheme requires the user to only scan a QR-code and verify the image that he/she receives on the smartphone which make it easy to use.

6. **Efficient to use:** A scheme is considered efficient to use, if the time spent on each authentication is short. CAPTCHA based schemes do not necessarily offer this benefit because the time needed for generating and displaying the graphical CAPTCHA and then input by the user via mouse is much more as compared to other schemes. QR-code based scheme proposed by Xie et al. [16] also does not offer this benefit because in addition to username and password, it requires 2 QR-code scans which increases the login time. The proposed scheme does not require the user to enter any credentials and only requires 1 QR-code scan and the complete authentication process takes an average time of 4.744 seconds which is less than many OTP/PIN, CAPTCHA, QR-code and hardware token based schemes. Thus, the proposed authentication protocol completely offers this benefit.

7. **Infrequent error:** An authentication scheme does not offer this benefit if a legitimate user is routinely rejected or the scheme is not reliable. Most of the schemes partially offers this benefit such as OTP/PIN based schemes may generate and send an incorrect OTP to the user. Schemes proposed by Xie et al. [16], Kim et al. [17] and Leung et al. [20] do not offer this benefit because these schemes involve more than one barcode scan, matching of

geolocation based on IP addresses and generation of graphical CAPTCHA respectively and these techniques can have frequent errors. The proposed scheme involves only one QR-code scan containing the session token in which errors are very infrequent and thus, the proposed scheme completely offers this benefit.

8. **Easy Recovery:** Most of the authentication schemes offer this benefit either partially or completely. If the user forgets the credentials or any token is lost and the user account is recovered easily, then the scheme offers this benefit. However, some schemes which involve a smartphone or any external device may require some extra steps for user verification and registering the device again. Due to this reason, the proposed scheme partially offers this benefit.

9. **Accessible:** If an authentication scheme can be accessed by all the users including those having some physical condition or disability, then the scheme offers this benefit. Also, if the scheme requires the user to have some technical knowledge particular to that scheme, then it cannot be considered as accessible. CAPTCHA based scheme are not accessible because the users having visual impairment cannot use a scheme which involves graphical passwords. Similarly QR-code based schemes are also not accessible as scanning a QR-code requires visual interaction and therefore, the proposed scheme is not accessible.

10. **Negligible cost per user:** OTP/PIN or an external hardware token adds a certain amount of cost to an authentication scheme and thus these schemes do not offer this benefit. However, the need for internet in a smartphone does not add any cost per user because internet has now become synonymous with smartphones. This is why the proposed scheme completely offers this benefit.

11. **Server compatible:** A scheme is server compatible if it doesn't need any separate implementation on the server side. Generating QR-code, certificates, graphical CAPTCHA or sending and receiving information from the Android app requires special implementation on the server side. Requirement of any specific platform on the server side also makes the scheme server incompatible. Most of the schemes are not server compatible as they enhance the security of the scheme by additional implementation on the server side. U-PWD and the

password managers are server compatible as they do not require any separate implementation on the server side.

12. **Browser compatible:** If the user needs to install any specific module to the browser or the scheme requires specific version of HTML or any other scripting language, then the authentication scheme cannot be considered as browser compatible. This means that a browser compatible scheme does not require any additional software or module to be installed in the client's machine. Schemes that require an extension on their browser to be installed in thus considered as Quasi-Browser compatible. Since the proposed scheme does not have any client or browser side requirement, therefore it is browser compatible.

13. **Mature:** A scheme is considered mature if it is rigorously tested and has been deployed for people. Quasi-mature is granted to those schemes which have only slight variations from the existing authentication schemes. QR-code based schemes proposed by Mukhopadhyay et al. [18] and Dodson et al. [19] are similar to WhatsApp web authentication [33], Google 2-step [15], password managers [23], Yahoo push login [22] and hardware token based schemes [25, 26, 27] exist in open view for public use and thus they are mature. The proposed scheme has slight variations which can be easily put to use and thus it partially offers this benefit.

14. **Non-proprietary:** If a scheme can be used without any explicit consent from the author of that scheme, then the scheme is considered as non-proprietary. Schemes such as Google 2-step [15] and hardware token based schemes does not reveal the complete information regarding the generation of OTP or token and thus they cannot be considered as non-proprietary. The proposed scheme can be used by any person without any approval and thus it is non-proprietary.

15. **Resilient to physical observation:** A scheme is safe from physical observation attack if the attacker is unable to obtain all the credentials required to login into the website. Shoulder-surfing and observing the keyboard comes under the category of physical observation. The basic U-PWD scheme is not secure against this attack whereas schemes such as Google 2-step [15], SAASPASS [14], password managers [23] and CAPTCHA based authentication

schemes [20, 21] partially offers this benefit because the attacker can obtain some credentials but not all the tokens to break the authentication. The proposed scheme is secure against this attack as the user does not enter any login details while authenticating into the website. The attacker won't get any login details by physically observing the user. Moreover, the secret key is stored in the Android Keystore API [35] which is not visible and thus, the attacker won't be able to login into the website by physical observation.

16. **Resilient to Target impersonation:** If an authentication scheme breaks when the attacker masquerades as the user to access his/her account, then it is not resilient to target impersonation. U-PWD and password managers are vulnerable to target impersonation as the attacker can get access to the account using recovery option with the help of the personal information of the user and thus they do not offer this benefit. The proposed scheme, OTP/PIN based schemes, QR-code, CAPTCHA and hardware token based schemes are resilient to target impersonation because some credential/token or secret key is not available with the attacker and thus, the attacker is not able to break these authentication schemes by impersonating the target.

17. **Resilient to Throttled guessing:** If the number of guesses has been restricted to a particular number by the verifier, then it is known as throttled guessing. If an authentication scheme is compromised by the attacker within the limit of the guesses allowed by the verifier, then the scheme is not resilient to throttled guessing. U-PWD and password managers are not resilient to throttled guessing because the attacker can easily guess a user's weak password. The proposed scheme along with other schemes taken into comparison are resilient to throttled guessing because all these schemes have some token that the attacker cannot know just by mere guessing.

18. **Resilient to Unthrottled guessing:** In unthrottled guessing, the verifier does not put any limit on the number of guesses. However, the number of guesses are limited by the computing resources available to the attacker. Just like the throttled guessing, the basic U-PWD and password managers are not resilient to unthrottled guessing whereas all other schemes completely offer this benefit because they are safe from this attack.

19. **Resilient to Internal observation:** Most of the authentication schemes are still vulnerable to the internal observation in which the attacker obtains user's credentials by using tools like keylogging or by using malicious browser extensions. Since the proposed scheme stored the secret key in the smartphone which is encrypted by Android Keystore API, therefore the attacker is not able to obtain any token. Thus, the proposed scheme is resilient to internal observation.

20. **Resilient to leaks from other verifiers:** The schemes which involve third-party for verification or other purposes are not resilient to leaks from other verifiers. Since the proposed scheme does not have any third-party verifier, therefore it secure against this attack.

21. **Resilient to Phishing:** If a scheme is safe from RT MITM, CR MITM and traditional phishing attacks, then it is resilient to phishing. The proposed scheme is resilient to phishing because even after relaying the QR-code to the phishing website which is opened by the user, the attacker cannot access the user's webcam due to which image will be taken by the attacker's webcam and not that of user. The analysis of all the attacks considered has been given in detail in section 4.3.2.

22. **Resilient to theft:** If the smartphone is stolen by the attacker, then it can be used for logging into the website. However, the smartphones are password or pin protected which makes it difficult for the attacker to access the smartphone app. Moreover, the attacker would still not be able to access the secret key as it stored secure using Android Keystore API [35]. Also, the user will notify the verifier about the stolen device via email which will lead to blocking of the account and the secret key stored in the device would be expired and the user will be asked to register again. Thus, the proposed scheme is Quasi-Resilient-to-theft. Dodson et al. [19] and password managers are not resilient to theft as the attacker can easily get access to the user account after stealing the smartphone/PC.

23. **No trusted third party:** The schemes having trusted third party for verification or generating the secret key or credentials do not offer this benefit. Due to this reason

Mukhopadhyay et al. [18], Tricipher [25], SAASPASS [14] and password managers [23] do not offer this benefit.

24. **Explicit consent:** The schemes which require explicit consent from the user before logging into the account offers this benefit. Except password managers, all other authentication schemes need consent from the user explicitly because user's credentials are entered automatically by the password managers.

25. **Unlinkable:** If the attacker can determine that the same user is trying to login into different websites, then the scheme is linkable. Most of the schemes are unlinkable except Kim et al. [17] because IP address is used as a token in the scheme proposed by Kim et al. [17]. When the user will login into different websites using the scheme proposed by Kim et al. [17], then it can determined from the IP address that the same user is accessing accounts on different websites. The proposed scheme and all other schemes offer this benefit completely as there is no such token used in the authentication scheme which can be used for linking. Table 4.5 shows the comparison of the authentication schemes based on Bonneau et al. framework [28]. From the table, it can be observed that the proposed scheme obtained a count of 18 out of 25 which is more than all other scheme. Thus, it can be concluded that the proposed scheme performs better than other schemes in terms of usability, security and deployability.

CHAPTER 5

CONCLUSION AND FUTURE WORK

3.1 Conclusion

In this thesis, an improved authentication scheme has been proposed which is able to handle traditional as well as advanced phishing attacks. Not only the proposed authentication protocol is able to handle RT MITM and CR MITM attacks, but it also safeguard the user from malicious browser extensions and app spoofing. Additionally, the proposed scheme offers the following benefits:

a) The proposed scheme does not require the users to remember any token or credential, as they just have to scan the QR-code and verify the image taken from the webcam. This makes the proposed scheme more user-friendly as compared to other existing authentication schemes.
b) The users are not required to carry any external hardware token or install Bluetooth for the purpose of authentication.
c) It is more intuitive than other schemes such as OTP based, QR-code based, hardware token based schemes or schemes requiring the user to scan and pair Bluetooth device.
d) The proposed scheme is platform-independent as it is compatible with popular platforms and browsers.
e) It does not require the user to install any additional software for the purpose of authentication.
f) The complete authentication procedure is automated and the user-interaction is minimal as he/she does not have to enter any part of the credentials.
g) It does not depend on any third-party verifiers or metrics that can be spoofed.
h) Unlike Tricipher [25] scheme, the proposed scheme is client-side independent.
i) The timing and resource analysis of the proposed scheme shows favorable performance.

j) The proposed authentication scheme performed better than other authentication schemes when they were compared in terms of usability, security and deployability.

3.2 Limitations

The proposed scheme requires internet on the user's smartphone and the user must have the smartphone with him for the purpose of authentication. However, the study [42] showed that the number of people having smartphones were 2.32 billion and this value will increase to 2.87 billion by 2020. Additionally, it has been shown in study [43] that approximately 80% of the people using internet have smartphones. Thus, internet has become synonymous with having a smartphone.

The proposed authentication protocol also requires a webcam on the client's terminal which is cheap and easily available. Moreover, most of the desktops nowadays have built-in webcam, especially laptops.

3.3 Future Work

The following work can be done in the future based on this thesis:

a) Currently, the proposed scheme requires the websites to use HTTPS for all communication and thus the proposed protocol can be implemented for secure cookie management so that it will not be vulnerable to session hijacking.
b) More extensive user testing of the proposed scheme can be done in the future to determine usability parameters such as learning-curve and scalability to see how the proposed scheme will handle simultaneous requests.
c) Tools such as Scyther, AVISPA, Spin, etc. can be used for the formal validation of the proposed protocol.

List of Publications

1. Ritwik Desai, Gaurav Varshney, Manoj Misra, Sajal Jindal, "Mobile App-Browser Extension Based Authentication", in *18th International Conference on Security and Management (SAM 2019)*, Las Vegas, USA.

2. Sajal Jindal, Manoj Misra, "Multi-Factor Authentication for Anti-Phishing Using Mobile App and Webcam", in *First International Conference on Advanced Communication and Computational Technology (ICACCT-2019)*, Kurukshetra, India **(submitted).**

REFERENCES

[1] E. Lastdrager, "Achieving a consensual definition of phishing based on systematic review of the literature," *Crime Science*, vol. 3, p. 9, 2014.

[2] T. N. Jagatic, N. A. Johnson, M. Jakobsson and F. Menczer, "Social phishing," *Communications of the ACM*, vol. 50, pp. 94-100, 2007.

[3] N. Abdelhamid, "Multi-label rules for phishing classification," *Applied Computing and Informatics*, vol. 11, pp. 29-46, 2015.

[4] M. Badra, S. El-Sawda, and I. Hajjeh, "Phishing attacks and solutions," in *Proceedings of the 3rd international conference on Mobile multimedia communications*, Nafpaktos, Greece, 2007, p.42.

[5] M. T. Banday and J. A. Qadri, "Phishing-A growing threat to e-commerce," *arXiv preprint arXiv: 1112.5732*, 2011.

[6] R. M. Mohammad, F. Thabtah, and L. McCluskey, "Tutorial and critical analysis of phishing websites methods," *computer science review*, vol. 17, pp. 1-24, 2015.

[7] DELLEMC. (2015). *Rsa online fraud resource center*. [Online] Available: http://www.emc.com/onlinefraud. [Accessed: 01 Apr. 2019]

[8] APWG. (2018). *APWG phishing attack trends reports.* Available: https://www.antiphishing.org/resources/apwg-reports/. [Accessed: 01 Apr. 2019]

[9] APWG. (2017). *APWG phishing attack trends reports.* Available: https://www.antiphishing.org/resources/apwg-reports/. [Accessed: 01 Apr. 2019]

[10] R. Dhamija and J. D. Tygar, "The battle against phishing: Dynamic Security Skins," in *Proceedings of the 2005 symposium on Usable privacy and security*, Pittsburgh, Pennsylvania, USA, 2005, pp. 77-88.

[11] G. Varshney, A. Sardana, and R. C. Joshi, "Secret information display based authentication technique towards preventing phishing attacks," in *Proccedings of the International Conference on Advances in Computing, Communications and Informatics*, Chennai, India, 2012, pp. 602-608.

[12] R. Islam and J. Abawajy, "A multi-tier phishing detection and filtering approach," *Journal of Network and Computer Applications*, vol. 36, pp. 324-335, 2013.

[13] Owasp. (2013). *Content Spoofing*. [Online] Available: https://www.owasp.org/index.php/Content_Spoofing. [Accessed: 01 Sept. 2018]

[14] I. Barker. (2015). *Saaspass makes two-factor authentication available to the masses*. [Online] Available: https://betanews.com/2015/01/15/saaspass-makes-two-factor-authentication- available-to-the-masses. [Accessed: 01 Aug. 2018]

[15] Google. (2015). *Stronger security for your google account*. [Online] Available: https://www.google.com/landing/2step. [Accessed: 01 Aug. 2018]

[16] M. Xie, Y. Li, K. Yoshigoe, R.Seker, and J. Bian, "CamAuth: Securing web authentication with camera," in *High Assurance Systems Engineering (HASE), 2015 IEEE 16th International Symposium on*. IEEE, 2015, pp. 232-239.

[17] S. H. Kim, D. Choi, S. H. Jin, and S. H. Lee, "Geo-Location based QR-Code authentication scheme to defeat active real-time phishing attack," in *Proceedings of the 2013 ACM workshop on Digital identity management*, Berlin, Germany, 2013, pp. 51-62.

[18] S. Mukhopadhyay and D. Argles, "An Anti-Phishing mechanism for single sign-on based on QR-code," in *Information Society (i-Society), 2011 International Conference on*, London, United Kingdom, 2011, pp. 505-508.

[19] B. Dodson, D. Sengupta, D. Boneh, and M. S. Lam, "Secure, consumer-friendly web authentication and payments with a phone," in *International Conference on Mobile Computing Applications, and Services*, Santa Clara, CA, USA, 2010, pp.17-38.

[20] C.-M. Leung, "Depress phishing by CAPTCHA with OTP," in *Anti-counterfeiting, Security, and Identification in Communication, 2009. ASID 2009. 3rd International Conference on*, Hong Kong, China, 2009, pp. 187-192.

[21] B. B. Zhu, J. Yan, G. Bao, M. Yang, and N. Xu, "Captcha as Graphical Passwords-A New Security Primitive Based on Hard AI Problems," *IEEE transactions on information forensics and security*, vol. 9, pp. 891-904, 2014.

[22] Yahoo. (2016). *Yahoo Sign In*. [Online] Available: https://login.yahoo.com/. [Accessed: 01 Aug. 2018]

[23] Lastpass (2017). *Lastpass*. [Online] Available: https://www.lastpass.com/. [Accessed: 01 Aug. 2018]

[24] B. Ross, C. Jackson, N. Miyake, D. Boneh, and J.C. Mitchell, "Stronger Password Authentication Using Browser Extensions," in *Usenix security*, Baltimore, MD, USA (2005), pp. 17-32.

[25] Tricipher, "White paper preventing man in the middle phishing attacks with multi-factor authentication", [Online]. Available: https://www.globaltrust.it/documents/press/phishing/PhishingSolutionWhitepaper.pdf [Accessed: 01 Aug. 2018]

[26] Yubico. (2017). *Fido u2f (universal 2nd factor)*. [Online] Available: https://www.yubico.com/about/background/fido/. [Accessed: 01 Aug. 2018]

[27] RSA. (2002). *RSA SecurID.* [Online] Available: https://www.rsa.com/en-us/products/rsa-securid-suite/rsa-securid-access/. [Accessed: 01 Aug. 2018].

[28] J. Bonneau, C. Herley, P. C. Van Oorschot, and F. Stajano, "The quest to replace passwords: A framework for comparative evaluation of web authentication schemes," in *Security and Privacy (SP), 2012 IEEE Symposium on*, San Francisco, California, 2012, pp. 553-567.

[29] OWASP. (2017). *Qrljacking*. [Online] Available: https://www.owasp.org/index.php/Qrljacking. [Accessed: 01 Oct. 2018]

[30] M. M. P. Center. (2013). *Browser extension hijacks Facebook profiles.* Available: https://blogs.technet.microsoft.com/mmpc/2013/05/10/browser-extension-hijacks-facebook-profiles/

[31] H. Shahriar, K. Weldemariam, M. Zulkernine, and T. Lutellier, "Effective detection of vulnerable and malicious browser extensions," *Computer & Security*, vol. 47, pp. 66-84, 2014.

[32] V. Gaurav, M. Manoj, K. A. Pradeep, "Detecting and Spying Fraud Browser Extensions: Short Paper," in *Proceedings of the 2017 on Multimedia Privacy and Security*, Dallas, Texas, USA, 2017, pp. 45-52.

[33] WhatsApp Inc. (2015). *WhatsApp Web.* [Online] Available: https://web.whatsapp.com/. [Accessed: 01 Oct. 2018]

[34] L. Malisa, K. Kostiainen, and S. Capkun, "Detecting mobile application spoofing attacks by leveraging user visual similarity perception," in *Procedings of the Seventh ACM on*

Conference on Data and Application Security and Privacy, Scottsdale, AZ, USA, 2017, pp. 289-300.

[35] Android Developers, "Android Keystore System," [Online]. Available: https://developer.android.com/training/articles/keystore [Accessed: 01-Dec-2018].

[36] Firebase. (2016). *Firebase Cloud Messaging.* [Online] Available: https://firebase.google.com/docs/cloud-messaging. [Accessed: 15 Feb. 2019]

[37] S. Owen, D. Switkin, Zxing Team. (March, 2008). *Zxing.* [Online] Available: https://github.com/zxing/zxing. [Accessed: 01 Feb. 2019]

[38] Twilio. (February, 2010). *Twilio.* [Online] Available: https://www.twilio.com/. [Accessed: 15 Jan. 2019]

[39] B. Firyn. (2012). *Webcam Capture.* [Online] Available: https://github.com/sarxos/webcam-capture. [Accessed: 01 Feb. 2019]

[40] J. Bonneau and S. R. Preibusch, "The Password Thicket: Technical and Market Failures in Human Authentication on the Web," in *WEIS*, Harvard University, USA, 2010, pp. 1-48.

[41] G. Varshney, M. Misra, and P. K. Atrey, "Secure authentication scheme to thwart RT MITM, CR MITM and malicious browser extension based phishing attacks," *Journal of Information Security and Applications*, vol. 42, pp. 1-17, 2018.

[42] Statista. (2017). *Number of smartphone users worldwide from 2014 to 2020 (in billions).* [Online] Available: https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/. [Accessed: 01 Mar. 2019]

[43] R. Sukhraj. (2016). *38 Mobile Marketing Statistics to Help You Plan for 2017*. [Online] Available: https://www.impactbnd.com/blog/mobile-marketing-statistics. [Accessed: 01 Mar. 2019]

[44] D. Wang, and P. Wang, "Two Birds with One Stone: Two-Factor Authentication with Security Beyond Conventional Bound," *IEEE Transactions on Dependable and Secure Computing*, vol. 15, pp. 708-722, 2018.

[45] G. Varshney, M. Misra, "Push notification based login using BLE devices," in *2nd International conferences on Information Technology, Information Systems and Electrical Engineering (ICITISEE)*, pp. 479–484, 2017.

[46] Ritwik Desai, Gaurav Varshney, Manoj Misra, Sajal Jindal, "Mobile App-Browser Extension Based Authentication", in *18th International Conference on Security and Management (SAM 2019)*, Las Vegas, USA.