FBNAuth: Secure MFA Scheme To Thwart Web-Based Phishing Attacks

Sneha Shukla, Gaurav Varshney
Department of CSE, IIT Jammu
sshukla19.01.02@gmail.com, gaurav.varshney@iitjammu.ac.in

***Abstract*—Increase in security breaches in the past decade manifests the security problems of web authentication using a single factor such as passwords/PIN. Thus, securing user credentials against web-based phishing attacks is a need of an hour. Despite being more secure and strongly promoted, two-factor (2FA) or multi-factor (MFA) schemes either fails to protect against recent phishing threats such as real time MITM, control relay MITM, malicious browser extensions based phishing attacks and/or need the users purchase and carry other hardware (security keys or tokens) for additional account protection. Leveraging the unprecedented popularity of NFC and BLE Enabled smartphones, we explore a new horizon to designing a MFA scheme for anti-phishing. This paper presents FBNAuth, a web authentication scheme that uses Bluetooth address as a device identification token. In FBNAuth, a mobile device is used as a multi-authentication factor to vouch for the identity of a user who is performing a web login from a mobile/PC. We have implemented a prototype system of FBNAuth that consists of a BLE-NFC-enabled android device. The threat modeling of the scheme suggests that it is safe against known phishing attacks. The scheme has been compared with other popular schemes using the Bonneau et al. assessment framework in terms of usability, deployability, and security. The results obtained are encouraging and suggests that the scheme can be used as an alternative to existing authentication schemes offering a similar level of security.**

***Index Terms*—Phishing, RT/CR MITM, Malicious Browser Extension, Bluetooth, NFC, Web Authentication, Keyloggers**

I. INTRODUCTION

Phishing attacks continue to play a dominant role in the digital threat landscape, as mentioned by APWG (Anti-Phishing Working Group), 2020 [1]. In traditional phishing attacks, attackers lure victims and make them input their credentials on a look alike sign-in page of a popular site that they have targeted. However, this technique became less efficient with the adoption of the two-factor authentication (2FA) [2] [3] scheme, which requires two authentication factors for successful authentication. Soon after the adoption of 2FA a variety of new techniques bypassing [4] the 2FA have emerged, with some of them leveraging the MITM (Man-In-The-Middle) phishing attack scenarios such as RT (Real Time)/CR (Control Relay) MITM phishing attacks [5] [6]. While 2FA provides another layer of security, it mainly protects against static or long term credentials (majorly username/password) stealing attacks. Addition of a second dynamic factor such as an OTP thwarts any attacks where attacker has an access to a static credential of a user. It does not protect against other types of phishing attacks like MBE (malicious browser extension) [7] [8]. These malware extensions often have legitimate functionality, but they act as Man-in-the-Middle to steal the data, this includes capturing of any second-factor which is used for login. Any authentication scheme is susceptible to a compromise if all credentials (whether dynamic or static) used for verification of an identity are entered by the user and has a proven trivial way to be captured by attackers in real-time.

Existing coveted web-based authentication schemes are likely to hold the above conditions. Therefore, the attacker could steal the identity information and access the user's sensitive information. The new authentication method presented in this paper is a multi-factor, and an innovative way to authenticate users by using the user's real-time face biometric identity (inherence factor) and BLE-NFC enabled mobile devices (ownership factor). The pairing of two BLE enabled devices is required to associate them before the final authentication. Using associated devices is a robust way to prevent web-based phishing attacks on authentication protocols in real-time.

Throughout this paper, the term "first device" refers to the mobile device attempting to authenticate some system and is registered device. The term "second device" refers to device on which login is being performed for successful authentication. Also, the "challenge" term represents a required action to complete authentication. The authentication process is performed by using two smart-phones. Also, as a challenge, a Nonce and Bluetooth address is used. When the first device tries to login into second device, the first device captures user's real-time face biometric using the MobileFaceNet [9] model employs less than a million parameters and is specifically modeled for high-accuracy real-time face verification on mobile device and generate real-time dynamic face-bio URL for login phase. Then user needs to perform NFC-tap from first device against second device to transfer token which is then verified by server, that accesses the same token from second device before verification. Finally the second device checks whether the first device is in it's proximity to authenticate the identity of first device. To test the working prototype two BLE-NFC enabled smartphones, and an Android application was developed. The main contribution of the paper is outlined below:

1) The security of existing multi-factor authentication schemes has been analyzed against the latest attacks that an attacker carries out via RT/CR MITM, and

MBE-based phishing. Our analysis indicates that most of the popular schemes are not capable of handling these attacks. Only a handful of them, such as Yubikey U2F, can mitigate these attacks, but they require a user to purchase a hardware key/token and always carry additional hardware with them.

2) We propose a secure password less authentication scheme FBNAuth that can handle RT MITM, CR MITM and MBE based phishing attacks and uses a BLE-NFC enabled smartphone, a device generally owned by every user using the Internet.
3) Using an automated security protocol verification tool, the proposed authentication protocol is verified.

The remainder of the article is organized as follows: Section 1 introduces the phishing attacks trending on the web and establishes the motivation for research in this area. It also outlines the main contributions of the paper. Section 2 describes the popular and existing multi-factor authentication schemes widely used for web authentication and highlight those proposals that are claimed to be capable of handling RT MITM, CR MITM, and MBE-based phishing attacks. The section also lists the identified research gaps. Section 3 describes the design and working methodology of the proposed secure password-less authentication scheme that addresses the identified research gaps in section 2. Section 4 provides details of the implementation, performance and security evaluation, and shows a comparison with the existing multi-factor schemes in terms of usability, deployability, and security. The authentication scheme was verified using an automated security protocol verification tool. As an initial step, the protocol was modeled using 4a high-level protocol specification language to verify and check whether secrecy and authenticity properties were violated or not. Section 5 discusses the key limitations of the proposed model and concludes with descriptions of the future work possible in the area.

II. RELATED WORK

In this section, we particularly review 2FA mechanisms, face recognition and bluetooth based - proximity authentication schemes, that are proposed to handle the recent phishing attacks

Face recognition using mobile: In 2015, Xie et al. [10] proposed CamAuth, a 2FA scheme that leverages user's mobile as a second authentication factor. The approach claims to solve the problem of MITM phishing attacks and utilizes a combination of Diffie-Hellman keys exchanged between the client browser and the server to prove the user's identity, and then verified using the user's mobile device via exploiting both the user PC and mobile cameras to exchange data that is encapsulated within a QR code. Phoolproof [11] is a public-key based scheme for strengthening bank transaction system. User is required to choose a bank site from the whitelist on the phone and then wait for information exchange between the phone and PC. The approach claims to thwarts Man-in-the-Middle attacks after setup, and protects a user's account even in the presence of keyloggers. In 2018, Zavrik et al. [1] proposed a 2FA web authentication scheme based on facial recognition to derive OTP and use this OTP to generate mobile device number ID for future login sessions. Recently, Xie et al. proposed CamTalk, a light based communication framework for bidirectional secure information transfer between smartphones by leveraging smartphone's display-camera channel [12].

Short-range communications: Bluetooth, WiFi, or NFC, are also widely adopted to support two-factor authentication. In 2019 [13], Ali et al. proposed web authentication using bluetooth devices that authenticate users by checking the presence of known or expected Bluetooth devices in the proximity of the user that can either be explicitly specified by the user, or it can be implicitly learned by the system through previous successful logins. Saxena et al. proposed a short-range device pairing protocol, VIC (Visual authentication based on Integrity Checking), which is also based on a unidirectional visual channel [14]. Another wireless communication channel (e.g., Bluetooth) has to be used to complete the pairing process. VIC is suitable for web authentication. An authentication service provider – SAASPASS [15] leverages on location-based iBeacon Bluetooth Low Energy (BLE) technology to authenticate users via Bluetooth communications between their registered phones and nearby login computers. Similarly, another 2FA proposal, PhoneAuth [16], sets up unpaired Bluetooth communications between a login computer and user's phone via Bluetooth using a new challenge-response protocol. However, these solutions may not be always applicable since most browsers (e.g., Firefox, Internet Explorer, and Safari [17]) do not support Bluetooth APIs. In addition, these solutions are not secure if adversaries set up Bluetooth connections to victims' phones to bypass 2FA. As NFC is widely embedded into today's commodity smartphones, Facebook [18] introduced a physical NFC security key that allows users to login to their accounts on their smartphones via NFC. This solution makes hardware token based two-factor authentication process faster. Instead of reading an authentication code from a hardware token and inputting it to a login computer, a user just taps a NFC security key against his/her smartphone so as to complete an authentication session. However, this solution requires additional hardware and its cost is of similar concern as in the case of hardware token based 2FA.

Hardware Token: Hardware token based 2FA is a widely deployed 2FA solution in practice (e.g., in financial industry). It requires users to carry and use hardware tokens for authentication. Other hardware security tokens such as RSA SecurID, during an authentication session, is used to generate an authentication code at fixed time intervals (usually 60 seconds) according to a built-in clock and a factory-encoded random key (known as "seed"). A user reads the authentication code from the hardware token and inputs it to a login computer after the user inputs the first factor. Hardware token based 2FA requires users to interact with their hardware tokens. It also requires a service provider to manufacture a number of hardware tokens and distribute them to all customers. In double armored Tricipher [19], multipart credentials are used.

In this scheme, one part of the credentials are stored in a secure enterprise data center, and the other is with the user. Also, the security key is stored on the user's device and is in the knowledge of the server. Both username and password entered by the user are encrypted using this key only. The enterprise data center encrypts (signs) user input with the part of the credential available on it and sends this encrypted message to the user. This encrypted message is then directly sent to the server, which completes the login process. The scheme needs an additional hardware security key during the login process. Yubikey uses the U2F protocol to authenticate the user during the web login. The Fast Identity Online (FIDO) Alliance was formed in 2012 by several companies (e.g., Lenovo, PayPal). The FIDO alliance aims to bring different authentication schemes together by providing a set of standards that simplify their adoption and use in web authentication. The FIDO website [20] has published its specifications recently. It aims to complete the verification of password in local device to avoid transmission of the password on the Internet, and use the dynamic data generated by the device as the authentication credentials to prevent attackers from compromising the user accounts by eavesdropping, phishing or other means. Since there is no static sensitive data dissemination on the web, login authentication security can be greatly improved. However, it can be seen that users do not have specific methods to manage their accounts in the FIDO service. The over-reliance on the device makes the user helpless when the device is unavailable. They will also feel impotent to manage the devices when users have more than one device. In contrast, our solution does not need any additional hardware except users' smartphones and it does not require users to interact with their smartphones in most cases.

III. THREAT MODEL

The attacker has the following set of capabilities:

1) **Phishing, MITM Phishing**: The attacker can trick victims into entering their credentials on the phishing website and use their credentials to gain access to their account. Additionally, a phisher can deploy remote desktop relay/ capturing modules (Ulterius, Teamviewer, etc.) on the victim machine or utilize QRLjacking [21] techniques to carry out RT MITM phishing and CR MITM phishing attacks.
2) **MBE Attacks**: The attacker can carry out this attack by installing a malicious browser-based extension, which provides a legitimate functionality in the foreground and performs stealthy activities such as logging user keystrokes, sniffing user-entered information by seeking the same set of user permissions for both foreground and background respectively.
3) **App Spoofing**: The attacker can install a spoofed Android App on the user's first device and can lure them into entering their account information over Apps [22].

IV. PROPOSED SCHEME: FBNAUTH

A. Assumptions

- Both the mobiles are assumed to have inbuilt BLE and NFC capabilities which are efficient than carrying an extra security key or token. The first device is NFC, and BT enabled and is in proximity to the second device.
- It is assumed that the user logs into the App installed in the second device using the App deployed on the first device.
- The user can use his first device (the one on which App is installed) as a registered device during the web login phase.
- The phisher is capable of the following :
  - The attacker can sniff BT_{ADDR_1} and install spoofed Android Apps and extensions during the android registration phase. The attacker can log users' keystroke with the help of MBE. It is assumed that the user will be using the legitimate Android App, and the web registration would be free from any attacks like many other schemes [10]
  - The scheme assumes that data communication between the browser (client) and the web server over HTTPS is safe from sniffing, and it can be used as a secure channel to exchange secret keys. Also, web servers and databases are assumed to be safe from any form of attacks.
  - The attacks carried out by hijacking sessions or by malformed DNS or by host-based malware is not in work's current scope, and it is assumed that the organization provides the secure DNS service. Also, the attacks launched due to a modified source code browser installed on the user's laptop are not in the current scope of the work.

B. Abbreviations used

Before presenting the FBNAuth system, we first introduce important FBNAuth notations for clarification purposes summarized in the Table I

C. FBNAuth System Overview

The proposed authentication scheme addresses some of the research gaps identified in the previous section. A user may log in on a website through his smartphone in the proposed scheme. A smartphone, and an Android App (or iPhone App) installed on it are required. The proposed multi-factor authentication scheme requires a real-time face biometric, device identification token (Bluetooth-address), and an instance id of the Android App. Unlike other schemes where the user enters his login username manually on the system, the Android App used in the proposed scheme captures real-time face biometric against active user session, generates a dynamic encrypted face-bio URL and a random secret Nonce "N1". The user needs to turn BLE on so that the first device can update both value N1 and Bluetooth address BT_{ADDR_1} of the first device to the database and sends this value to the web server

TABLE I
ABBREVIATIONS USED

Variable	Description	Function	Description
[-2.0ex] EM	Email Address	HTTPS(DATA)	HTTPS message carrying data
[-2.0ex] PWD	Password	GEN(x)	Generate a cryptographically secure random secret x.
[-2.0ex] $FBIO$	Real time face biometric	$CMP_{DB}(x, y, z....)$	Matches user credentials received as arguments with those present in the server database
[-2.0ex] $FBURL$	Real time face biometric URL	STORE(x)	Stores the value 'x',encrypted (symmetric encryption AES) with AWK and SALT in Android App storage.
[-2.0ex] $PBAPP$	Instance id of Android App	$SAVE_{DB}$	Adds a new tuple (EM,PWD,PBAPP, SALT,FBURL) to the server database.
[-2.0ex] AWK	Shared secret between device and server	REPLACE(a,b)	Replaces the encrypted value 'a' with encrypted value 'b' in server database storage
[-2.0ex] BT_{ADDRi}	Bluetooth Address of device i(login or registered device)	SEARCH(x)	Search the x value in the nearby proximity of the device using bluetooth technology
[-2.0ex] K_s	Secret to encrypt the challenges and decrypt the response	NFC(x)	Tap the encrypted token against the second device using NFC
[-2.0ex] $Nonce_1$	A random 10 digit long string generated for verification of registered device	$W_{DB}(x)$	Creates a new column in the server database and stores value 'x'

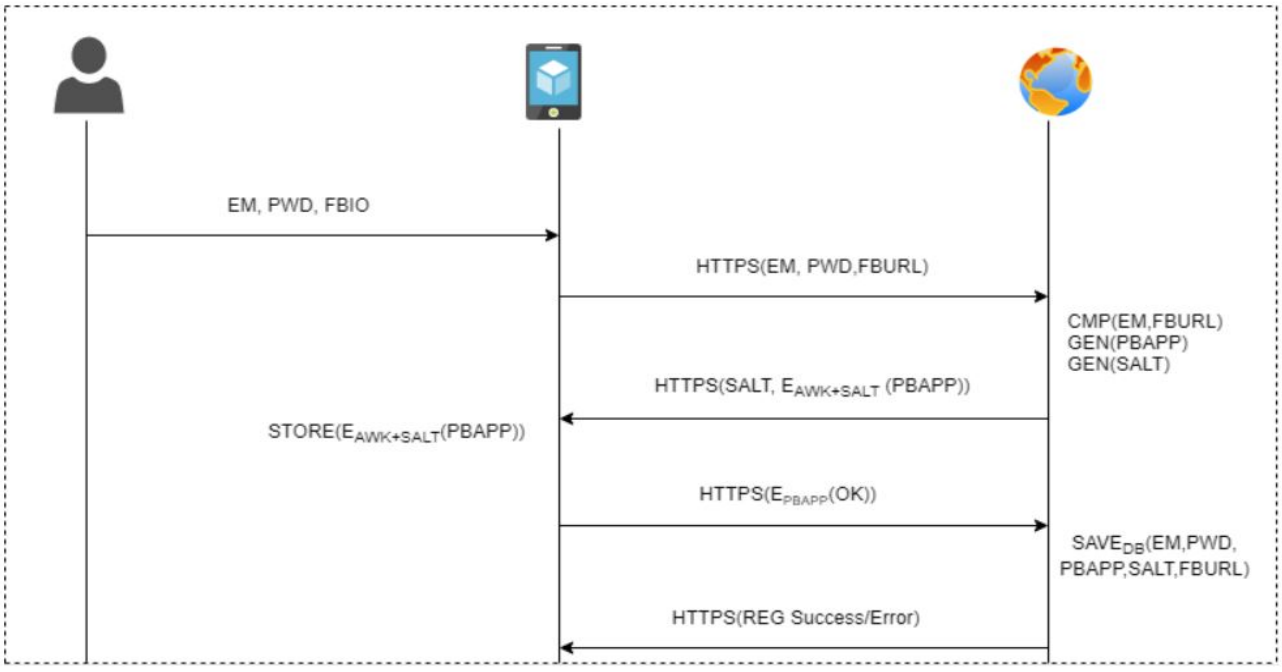

Fig. 1. Smartphone Registration Phase

over HTTPS for its identity validation as a first authentication factor. Once the server verifies the nonce. Subsequently, an encrypted token (image) containing face-bio URL, user's email address, and unique session-ID is generated as the second authentication factor. This increases the complexity of obtaining the face biometric during a phishing attack and makes it almost impossible for the attackers to receive this information from users via a reply to emails. The user needs to enable both NFC in both the devices and then perform an NFC tap from the first device against the second device to transfer the token (encrypted image). After the successful transfer of the token, the token gets deleted to avoid any misuse in the future. The user needs to tap the button that says "Login with NFC" using the Android App present on the second device to initiate the authentication process. The second device then fetches its Bluetooth address BT_{ADDR_2}, updates it to the database, and forwards it along with the received challenge to the server. The server verifies the received and updated BT_{ADDR_2} for the second device identity validation. Once the verification is successful, it decrypts the received token(image) to fetch the session ID, face-bio URL, and email-ID of the user trying to perform the login. The server then fetches the image from the face-bio URL and matches it with the image fetched from the registered face-bio URL. Suppose the match percent reaches a certain threshold, and the received session ID is valid. In that case, it updates the second device about the successful user identification, and the token gets deleted. The second device then starts searching for the BT_{ADDR_1} and compares it against the updated BT_{ADDR_1} in the previous step as the third authentication factor. If the match is successful, the user gains access to the login page during an authentication phase. From above, we can say that it is difficult to obtain, relay, and use all three information factors in real-time to perform an RT MITM attack, since the face-bio URL is stored inside a secret token (image) which is encrypted with a one-time secret key generated for every login session, which makes it difficult for an attacker to decrypts it and fetches the image in such a short span before it gets deleted from the first device. Also, it avoids the possibility of CR MITM, as the Bluetooth device remains physically paired with the user's second device. The user doesn't need to remember any authentication token during the login process and needs to perform only one tap to perform the account login. Mobile Phone App Instance Id (PBAPP), which is used along with SALT as a secret key to encrypt image files, is used for all the challenge encryption and is procured automatically by App.

D. FBNAuth Protocol Details

A user can log in to a website using the first device, which is a smartphone. The new user registers on the website during the registration phase for the first time. The proposed scheme includes two phases: (1) Android Registration Phase - first device (ARP - FD) and (2) Android Login Phase - second device (ALP - SD)

ARP - FD: In ARP - FD a new user registers on the website.

1) It's assumed that both android app on the first device and the web server has exchanged a shared secret key AWK using TLS/SSL handshake.
2) In ARP - FD user opens the android app on the first device and enters his personal information including the Email Id (EM), Password (PWD) as initial step of registration and clicks on the "Save" button. After entering their details, user provides his/her face biometric using android inbuilt camera app and clicks on "SignUp" button.
3) Once the user clicks on "SignUp" button, face-bio URL is generated and is sent along with personal information entered by user in step (3) to server over secure HTTPs channel
4) The server verifies EM uniqueness against existing users in the database. Once EM uniqueness is verified, server generates unique as well as secret PBAPP, SALT and $E_{AWK+SALT}(PBAPP)$ and sends SALT and $E_{AWK+SALT}(PBAPP)$ to Android app over HTTPS connection.
5) The Android App stores $E_{AWK+SALT}(PBAPP)$ in it's local storage
6) The Android App sends back $E_{PBAPP}(OK)$ to confirm that the $E_{AWK+SALT}(PBAPP)$ has been saved successfully in its storage. If the server receives a valid $E_{PBAPP}(OK)$ it creates a database entry (REGDB) and sends a successful registration message, otherwise, it sends an error message.
7) Face- Biometric and SALT are not stored in the Android App. The registration phase happens only once for every user.

The registration on the first device is not the main thesis contribution. Hence, the security analysis of the android registration phase is not addressed in the thesis. The messages exchanged during ARP - FD is shown in Fig 1.

ALP - SD: To login into his registered account on the second device user must be logged in to his SP App account on the first device. Logging over the SP App is needed for this authentication model to take real-time face bio-metric and transfer challenge to the second device through NFC tap to log into the second device.

Now user should follow the below steps to login into his registered account using the second device:

1) In this phase, unlike conventional authentication schemes, the the user only takes real-time face biometric from the Android App login screen opened on his registered smartphone (first device).
2) Once user provides their face bio-metric, app will generate a secret random number Nonce N_1 and obtains BT_{ADDR_1} of first device automatically and update (N_1, BT_{ADDR_1}) to database against the users registered EM address. Subsequently app initiates communication with the server and provides $D_{Nonce_1}(E_{AWK+SALT}(PBAPP))$ for the first device identification.

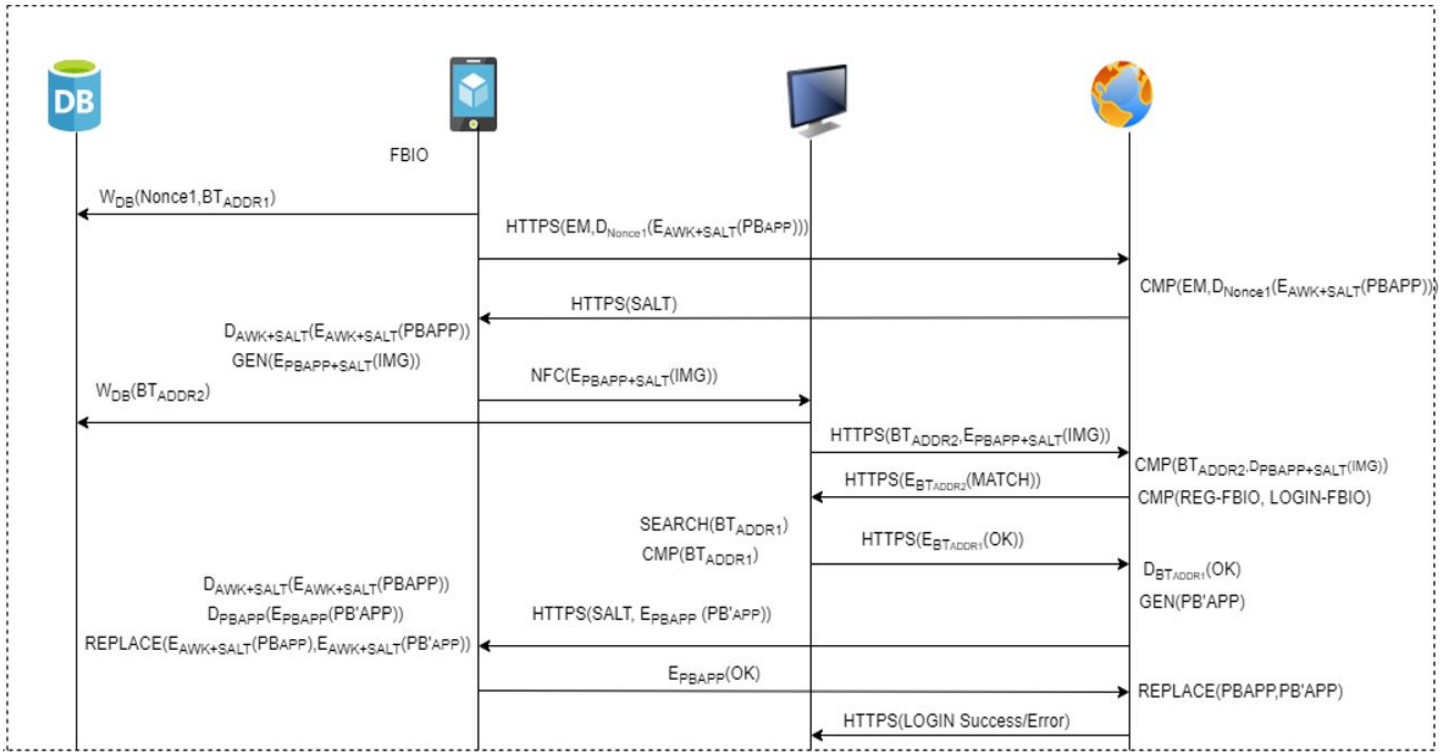

Fig. 2. Smartphone Login Phase

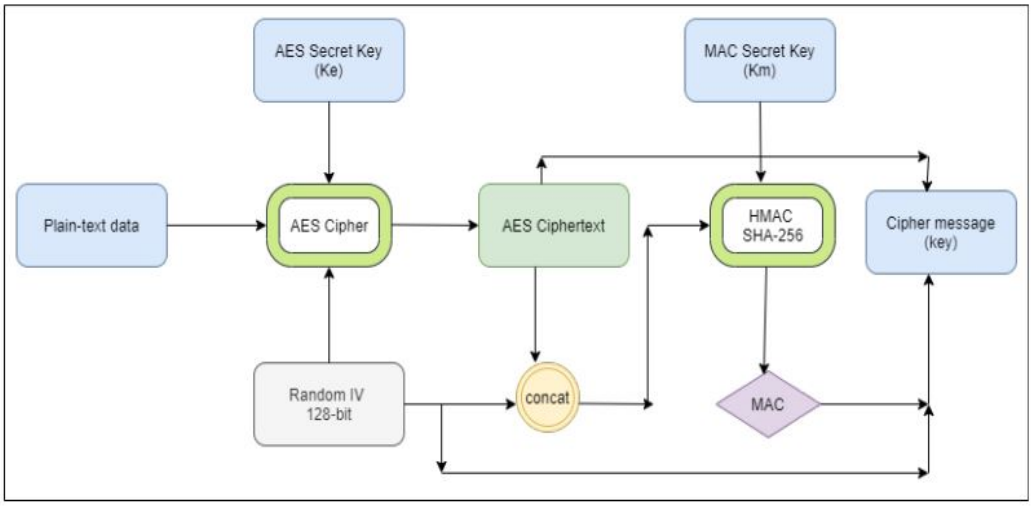

Fig. 3. Encrypt-then-MAC-scheme

3) The $D_{Nonce_1}(E_{AWK+SALT}(PBAPP))$ received from the first device is verified by the server to confirm the identity of the first device and ensure that the first device initiates SALT request over HTTPS. In order to verify this, the server encrypts the PBAPP stored in database with the Android $AWK + SALT$ and then decrypts it with N_1 to get the server copy of $D_{Nonce_1}(E_{AWK+SALT}(PBAPP))$.
4) If both the received and server copy of $D_{Nonce_1}(E_{AWK+SALT}(PBAPP))$ matches, then the server sends SALT to the first device.
5) The first device uses the received SALT from the server and uses first device secret key AWK to decrypt the PBAPP, locally stored in the first device. After getting PBAPP, it computes secret Key K_s using $SALT + PBAPP$.
6) Once secret key K_s is generated, it generates a secret challenge in form of an image encrypted with K_s and starts timer for the second device to complete the authentication before it expires.
7) User will select and share this secret image file containing EM, Face- Bio URL, session ID using App on the first device through NFC tap to the second device. After successful sharing, App on the first device purges this secret image from its local storage.
8) Now, the user will click on the "Sign In Using NFC" button on the second device. Then, the second device submits its BT_{ADDR2} to the database, initiates communication with the server, and provides both BT_{ADDR2} and the received encrypted image file. The server verifies BT_{ADDR2} for the identity verification of the second device.
9) Once the second device identity is verified, the server will derive session ID, EM, and image URL from the secret received. The server verifies an active session ID against the received EM address of the registered user. Once the session ID is verified, it compares both the images derived from registered URL and real-time face-biometric URL. If image-matching-value (IMV) crosses threshold, then it informs the second device about the successful match by sending $E_{BT_{ADDR2}}(MATCH)$ over HTTPS.
10) After this second device decrypts $E_{BT_{ADDR2}}(MATCH)$ with BT_{ADDR2} to obtain "MATCH" message. Then the second device starts searching for BT_{ADDR1} in its nearby proximity. Once the Bluetooth-enabled mobile device is found with BT_{ADDR1}, it sends $E_{BT_{ADDR1}}(OK)$ to the server. The server decrypts this received message using BT_{ADDR1} and generates a new PB'APP, and sends SALT $E_{PBAPP}(PB'APP)$ to the Android App on the first device.
11) Android app uses SALT from server and retrieves PBAPP by applying decryption algorithm $D_{AWK+SALT}(E_{AWK+SALT}(PBAPP))$ stored in the App local storage and uses this PBAPP to decrypt PB'APP by applying decryption algorithm $D_{PBAPP}(E_{PBAPP}(PB'APP))$ which can be used in next login session to generate K_s as $SALT + PB'APP$. It also replaces $E_{AWK+SALT}(PBAPP)$ with $E_{AWK+SALT}(PB'APP)$ in it's App local storage.
12) The Android App then uses the PBAPP of the current login session to generate $E_{PBAPP}(OK)$ and sends it to the server for key verification.
13) The server decrypts the challenge with the old PBAPP stored in its database and returns user account over HTTPS connection.
14) Finally, the server replaces PBAPP stored in its database with PB'APP

The messages exchanged during ARP - FD is shown in Fig 2. Appendix shows a video made during our testing.

E. PBAPP, Nonce, Ks and SALT generation

The PBAPP and Ks are some of the authenticators that are generated using the flutter library, which is cross-platform string encryption that uses AES256 CBC + PKCS5 + Random IVs + HMAC-SHA 256 to ensure confidentiality, integrity, and authentication of string. The Nonce is generated using the "random numeric" method, and SALT is generated using the "generateSalt" method of the same library.

In particular, consider this scenario to obtain the Key for encrypting any input string: Bob has an identifying key K, which is also shared with Alice, that he can identify himself with. Only both of them knows this key K. Bob then encrypts $(Nonce||K)$ using Alice's public key, and Alice decrypts it using its private Key to obtain Nonce, and K. Alice uses HMAC SHA-256 with $(K||Nonce)$ to yield K(e) of 256 bit and HMAC SHA-256 with $(K||Nonce + 1)$ to yield K(m) of 256 bits. To send a message to Bob, Alice needs to perform the following actions:

- A new random 128-bit Initialization Vector (IV) is created.
- The message is encrypted using IV and K(e) as the Key.
- A SHA-256 HMAC is created with K(m) as Key and $(IV||Encryptedmessage)$ as data.
- Then she finally sends $(IV||HMAC||Ciphertext)$ to Bob

The "generateKeyFromPassword" function from flutter is used to generate the key using the method described above, where user-provided input is plain-text and salt is IV. This generated key is used to encrypt any given input string (password etc) to obtain secure token (PBAPP/Ks) for our authentication model as shown in Figure 3 and prototype is shown in Figure 4

V. IMPLEMENTATION AND TEST SETUP

A. Test Setup

The following software and hardware were used for implementation and testing:

- A LG G8X ThinQ smartphone with Qualcomm SM8150 Snapdragon Chipset, 1.0 GHz Octa-core (1x2.84 GHz

```
Future<String> Encrypt(String password) async{
  var cryptor = PlatformStringCryptor();
  final salt = await cryptor.generateSalt();
  String key = await cryptor.generateKeyFromPassword(password, salt);
  String encryptedPassword = await cryptor.encrypt(password, key);
  return encryptedPassword;
}
```

Fig. 4. Encryption prototype

Kryo 485 3x2.42 GHz Kryo 485 4x1.78 GHz Kryo 485) CPU, Adreno 640 GPU, 128GB 6GB RAM, and Android 9.0 (Pie) operating system.

- The PC used for implementing the Android application was a Windows 10 machine with Chrome Browser 90.0.4430.212 Version and an Intel(R) CoreTM i5-1035G1 CPU @ 1.00 GHz 8.00 GB of RAM.
- For the testing registration phase, the smartphone login phase was hosted on a local PC running Windows 10 OS on an Intel(R) CoreTM i5-1035G1 CPU @ 1.00 GHz with 8.00 GB of RAM. The API scripts were written in PHP and MYSQL and were hosted on the XAMPP server.
- The first device hosting the application and the second device used for login were on the same LAN during the testing.
- Though the Android App has been used to implement the demo, the proposal can be implemented in other browsers and mobile operating systems.

B. Server Architecture

The primary process to implement the proposed model is the authenticator web server. A web application server is required to implement the verifier system of the authentication protocol. The web application server communicates with both the first and second devices via REST API . REST (Representational State Transfer) is a set of architectural constraints, not a protocol or standard for implementing any web services. The proposed design approach lets the web application server communicate with both the devices irrespective of their underlying operating system. The ”PHPMyAdmin Xampp” was used to implement the web server application. The android project was deployed and tested locally on a server, pre-installed with MYSQL database, Apache web server, Perl, and PHP to build an offline application with desired functionality.

C. Database Design

Another vital component of this protocol design is the device vetting process by using a Bluetooth address. To assess both the devices, user and device identification information was required. Considering that, a primary database prototype was designed and also implemented. An email address and face-bio URL were used to identify any user in real-time. Also, both the first and second device was authenticated using their Bluetooth hardware address against the registered user email address. The devices were associated for the first time during the login phase, and their Bluetooth addresses were updated the first time in the database during this phase only. The MySQL and flutter Real-time database were both used for data storage and data persistence. The Firebase Realtime Database is cloud-hosted. Data is stored as JSON and synchronized in real-time to every connected client. The Firebase database was used to store the user flutter auth ID, email address, and face-bio URL of the registered user. Subsequently, user details corresponding to this flutter auth ID are stored in the MYSQL database, ensuring a double layer of security.

D. Performance evaluation

The performance of our proposed model was gauged in terms of time taken for its various operations and the Memory and CPU utilization. ”Another Monitor” Android App was deployed over both the devices to record the Memory and CPU utilization as shown in Figure 5 and Figure 6 of our Android App ”FlyBuy.” We can see that the minimum CPU utilization of the FlyBuy Android App is 0.28% on the both devices, and the maximum CPU utilization was 4.80% on the both device. The memory utilization for first device was 12.2 MB and for second device 13.1 MB. The statistics confirm that the Memory and CPU utilization of our model is low enough, and it could be used for everyday login purposes. Fig 5 6 7 show the captured values. The memory utilization, CPU utilization, and login time of our model have not been compared with the other existing schemes because of the underlying reasons outlined below:

- The application login time is decided by two factors, i.e., the speed and availability of the web application server. The login time was captured when only one user tried to log in to the web server as shown in Fig. 7 Since; it is not feasible to record login time on a commercial website using other authentication schemes (User-PWD, Push notification based, QR Code) because they used to manage multiple customers over the same period.

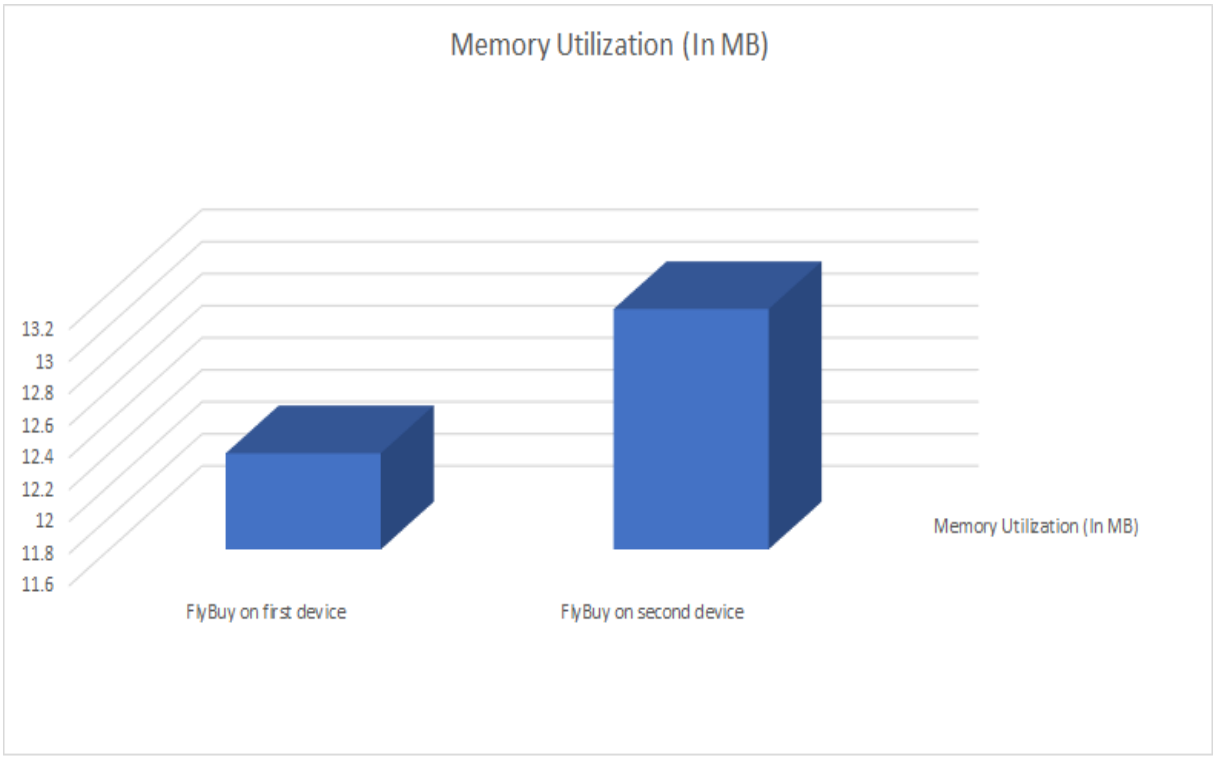

Fig. 5. Memory utilization: FlyBuy

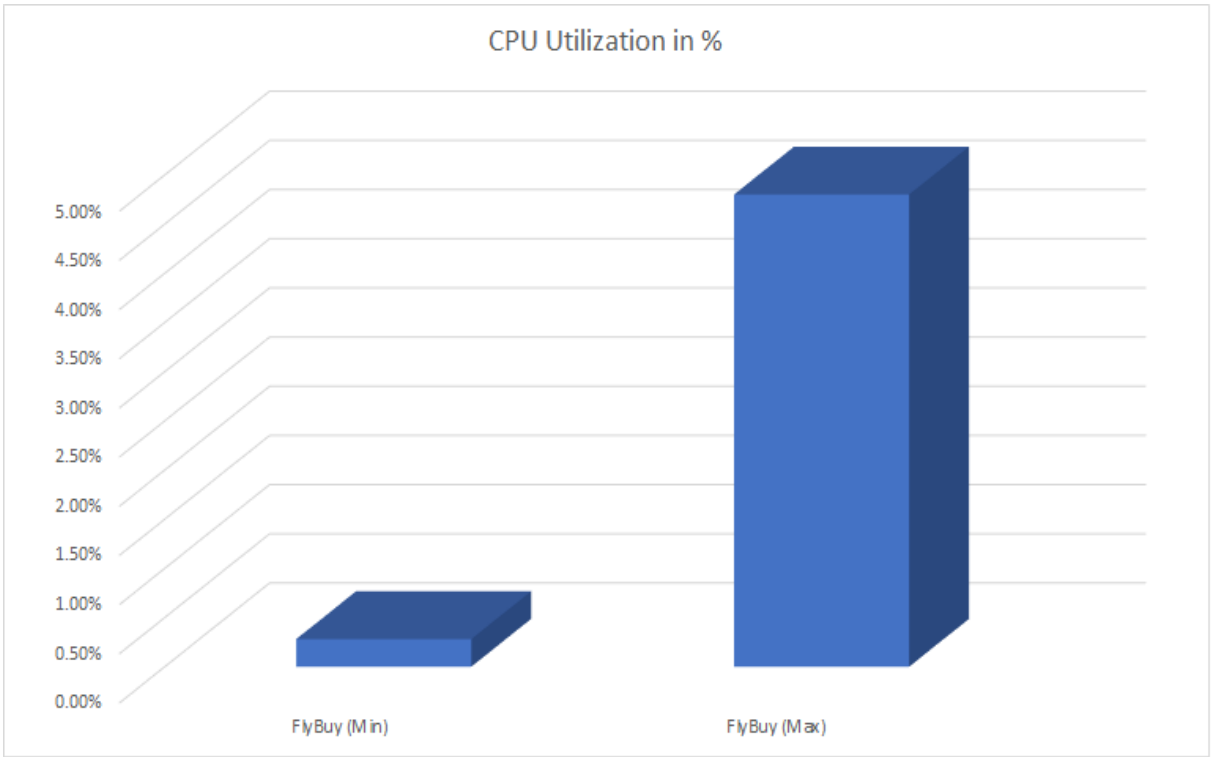

Fig. 6. CPU utilization: FlyBuy

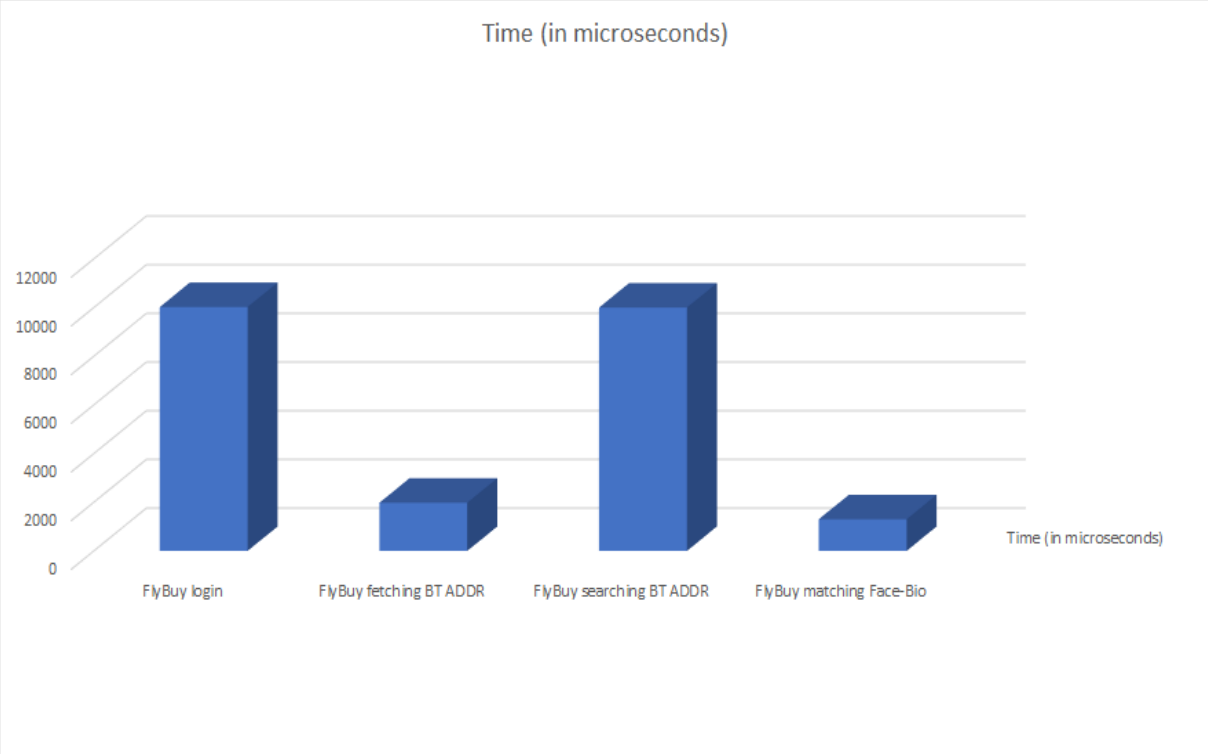

Fig. 7. Time needs through FlyBuy

- The Memory and CPU Utilization depends on the type of application supported by App. For example, an online social media app such as Instagram App consumes a lot of Memory and CPU because of the high media content they receive and process. Thus, a direct comparison of CPU and Memory Utilization is not a fair idea.
- Some of the non-commercial proposed schemes haven't mentioned anything related to Memory, CPU, and login time in the past.

We did numerous trials to record the additional average time for two parameters (1) when the first device tried to connect via Bluetooth technology with the second device and (2) when the first device tapped against the second device to transfer the token. Approximately 100 experiments were carried out, as shown in Figure 8 The average time values for the push notification, an extra time for graphical password entry, OTP delivery to a phone, and QR code scanning has been obtained from [23] respectively. Our experiment shows that the average time required for the first device to tap against the second device and connect two devices via Bluetooth technology takes less time than the schemes mentioned above.

VI. Security analysis

This section discusses the security of the proposed scheme against the attacks which can be carried out by an attacker to compromise the authentication scheme or steal the user credentials.

A. *Security against RT MITM phishing*

Device-specific bluetooth address BT_{ADDR_1} and BT_{ADDR_2} and user-specific $PBAPP$ cannot be acquired through remote desktop monitoring / remote screen relaying, malicious browser extensions or on phishing websites and hence cannot be relayed to an authentic website in real time to cause an RT MITM attack or a CR MITM phishing attack. The attacker will not be able to acquire BT_{ADDR_1} because this is automatically updated by first device to database server and second device detects the existence of BT_{ADDR_1} in it's proximity once update is done. Also, attacker can't obtain secret image file which is generated in encrypted form (encryption key: legitimate android app private key concatenated with server SALT) in the local storage of the authentic Android App and transferred to second device through NFC tap and deleted once transfer is done. The second device also deletes this secret file once login is completed or if the timer started by server expires during inactive session. Also, the attacker will need the $E_{AWK+SALT}(PBAPP)$ stored on the registered smartphone Android App.

B. *Security against CR MITM phishing*

CR MITM phishing is not possible because the use of BT_{ADDR_1} as the first device identification token and BT_{ADDR_2} as second device identification token since it is impossible to access the BTADDR of the first device connected to the user second device via Web Bluetooth APIs as Web

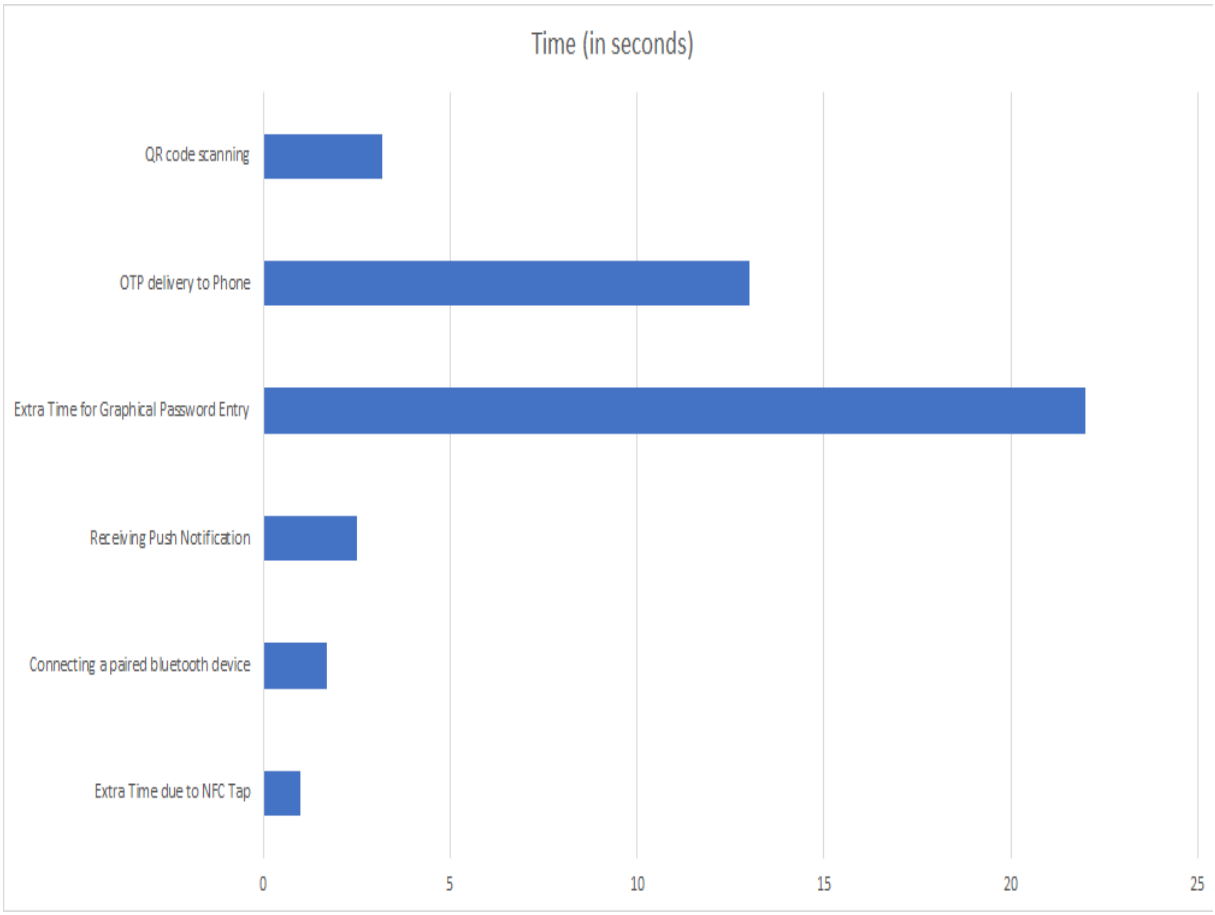

Fig. 8. Comparison of time needed for carrying out additional operations during authentication.

Bluetooth APIs can only access the BLE devices physically connected to the attacker's PC.

C. *Malicious Browser Extension based attacks*

As user, does not type any information on the website the credentials stealing attacks which happens through keystroke logging, PWD and HTML form data sniffing cannot happen.

D. *Host malware based keystroke logging*

Keystroke logging based attacks that steal credentials entered by the user over websites can also be avoided with the use of bluetooth address and NFC as the device identification token as BTADDR1 is automatically retrieved by the websites and secret is transferred to second device through NFC tap and is automatically retrieved by servers to verify user's biometric.

VII. MODEL CHECKING

A. *Model Checking : Overview*

This section highlights the modeling and verification of the authentication protocol. In order to verify that the web authentication protocol assures the secrecy and authenticity of communication between devices and the web server, a model checker must be used. Therefore, AVISPA [24] model checker was used to inspect secrecy and authenticity properties. The Automated Validation of Internet Security Protocols and Applications (AVISPA) tool provides a suite of applications for building and analyzing formal models of security protocols [24] for large-scale security protocols. The protocol models are written in the High-Level Protocol Specification Language, or HLPSL. The AVISPA Tool comprises four backends for backend verification [21]:

- OFMC (On-the-fly model checker)
- CL-AtSe (Constraint Logic-based Attack Searcher)
- SATMC (SAT-based Model-Checker)
- TA4SP (Tree-based model checker)

OFMC backend comes in handy for detecting, guessing, and carrying out replay attacks. The SATMC and CL-Atse backend platforms are generally used to check the bounded number of protocol falsification and sessions. TA4SP backend furnishes unbounded security protocol verification using tree-based languages [24]. T represents a server, Bob represents the first device, and Alice represents the second device for the proposed authentication model. Kat, Kbt, Kit is the keys commonly known between second device and server, first device and server, and intruder and the server, respectively. In AVISPA, there exist two security goals. In order to verify if the

TABLE II
MODEL CHECKER RESULT

Backend	Summary	Statistics
SAFE Analysed: 309673 states Reachable: 309673 states Translation: 0.01 seconds Computation: 31.05 seconds	parse time: 0.00s searchTime: 35.55s visitedNodes: 19136 nodes depth: 15 piles	OFMC SAFE CL-Atse

devices are authenticated to the server and to each other, and the first device Bluetooth address, the following goals were outlined:

1) Authentication on the first device
2) Authentication on the second device
3) Authentication on the webserver

Furthermore, in order to check if the communication was kept secret, the following goal was outlined:

1) Secrecy of secret key (Ks) (used for all token/data encryption)

B. Model Checking : Results

The Lenovo laptop was used during our experiments. The laptop computer is a Windows 10, which has 8.00 GB RAM, a 1.00 GHz Intel(R) CoreTM i5-1035G1 processor. The verification results are summarized in Table II. The CL-AtSe backend platform was used to verify the bounded number of sessions. CL-Atse completed the protocol verification in 31.05 seconds by analyzing 309673 states. As a result, this backend doesn't find any attack on the proposed protocol. In order to find replay and guessing attacks, the proposed model was verified by the OFMC backend. A heuristic search algorithm was run by OFMC with 15 piles and analyzed a total of 19136 nodes, and it was found SAFE from any of the possible attacks.

VIII. EXPERIMENTAL RESULTS

Fig 10 shows the registration screen using which user will create account. Then user can sign in user sign in screen as shown in Fig 11. Once user has logged in on the first device they can use the first device to login in the second device by taking picture using camera as shown in Fig 12. Once the image is captured a secret file is generated that is transferred to the second device using NFC tap, after which the second device will search for the first device proximity and allows user to login to the second device as shown in the Fig 13

IX. COMPARISON: USABILITY

In this section, above discussed authentication systems are compared in terms of tokens used by the system, the number of tokens need to be remembered, the number of taps required to send token, additional requirements such as the need of Internet on a smartphone for the model to work for Smartphone (second device) login. After comparison, it was found that most of the schemes only need a smartphone as an additional requirement which mostly Internet users carry every day. Other schemes need driver modules to be installed on PC, hardware token, GPS, trusted third party, etc. The requirement to have data or Wifi connection in the smartphone while logging in from a PC is a crucial factor in getting both the incurred cost using the proposed scheme and the issues that might arise due to the unavailability of internet connectivity on a smartphone. The comparative discussion has been outlined in Table III

X. ASSESSMENT WITH BONNEAU ET AL. FRAMEWORK

This section draws a comparison between the existing authentication schemes in terms of usability, deployability, and security using the Bonneau et al. framework and same can be seen in Figure 14, 15, 16 In the comparison table, some of the values are directly taken from the study [34], and some other values are modified according to the analysis performed on the scheme against the latest cyber-attacks. A brief description of different Bonneau et al.'s framework parameters have been provided, and benefits offered by existing schemes have been discussed.

1) **Usability**
   a) **Memory wise effortless** Nearly all the existing schemes are not memory-wise effortless (represented by) due to the fact that they want the user to remember username and password. Yahoo Push login [23], Password managers [33], Kim et al. [5] reduces the number of tokens to be memorized and input entered by the user and hence are categorized as scheme offering partially benefit (represented by) while our proposed scheme and Dodson et al.'s authentication scheme [28] doesn't require the user to enter any input on the website are considered as "offering the benefit" scheme (represented by).
   b) **Scalable for users** If login over multiple web accounts using a similar scheme increases the load on the user, then that authentication scheme is not a scalable one. Our scheme offers this benefit where the user needs to just connect the BT mobile with the PC once and have to log into their smartphone App before in order to log into their websites on PC.
   c) **Nothing and quasi-nothing to carry** It is considered that a user carrying his/her smartphone is almost equivalent to carrying nothing. Our scheme offers this benefit given the fact the nearly every internet user carries a smartphone. Rather if the user is required to carry an additional device or

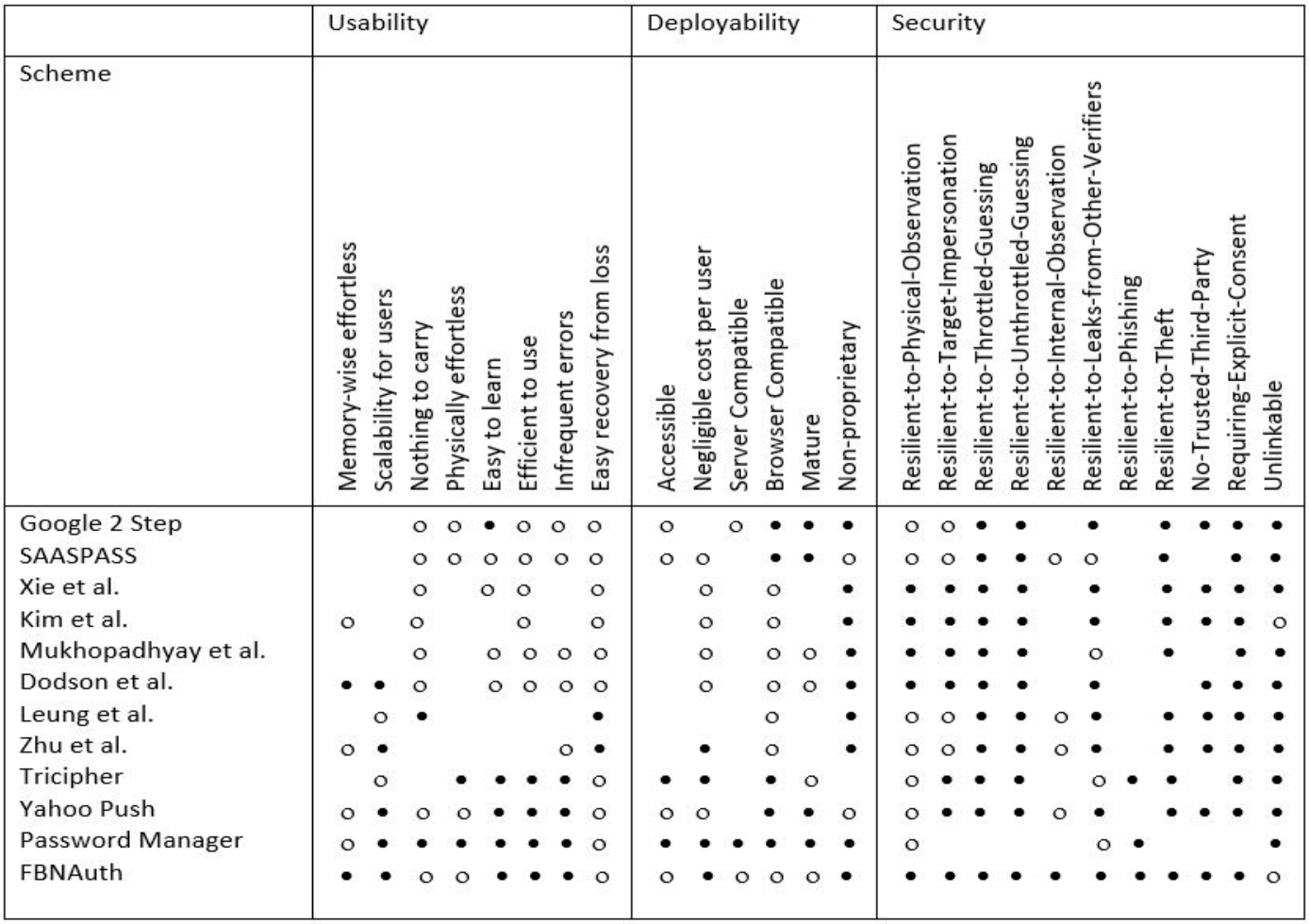

Fig. 9. FBNAuth Scheme Evaluation indicates that the scheme fully carries the characteristic indicates that the scheme partially carries the characteristic (the Quasi prefix). We take rows 1-11 from

hardware, then the scheme doesn't offer this benefit. For example, Tricipher [30] login is client-system dependent, due to which it doesn't offer this benefit.

d) **Physically effortless** If the user doesn't need to perform extra efforts such as tapping a button or entering credentials, then the scheme is considered to be physically effortless. There are so many existing schemes that do not offer this benefit completely, as the user might have to fetch a PIN or OTP in order to interact with the smartphone. Some schemes require the user to perform more interaction with the device, such as scanning Barcode or CAPTCHA entry. Our proposed model is rationally physical effortless since NFC tap, or Bluetooth pairing or capture Image using Camera is a one-click process.

e) **Easy to learn** Any scheme which user can learn with the basic knowledge of computer and uses the scheme, then that scheme is considered to be easy to learn. The schemes such as CAPTCHA or graphical passwords, QR code scanning don't offer this benefit mostly. However, the proposed scheme is easy to understand and learn since the user only needs to know how to do BT pairing of the devices. Also, BT pairing is done once between two devices by means of a one-click process. Even people belonging to the old age group lacking technical knowledge can be guided once to perform BT pairing, and after that, they can simply connect with PC at any time of the day.

f) **Efficient to use** If the login time is minuscule, then the authentication scheme is considered to be an efficient one to be used. The existing schemes such as QR code-based schemes OTP based schemes offer this benefit partially only, while schemes such as graphical password do not offer this benefit at all due to the time it takes to generate and show CAPTCHAs, other than the time taken by any user to enter it using mouse clicks. Our proposed

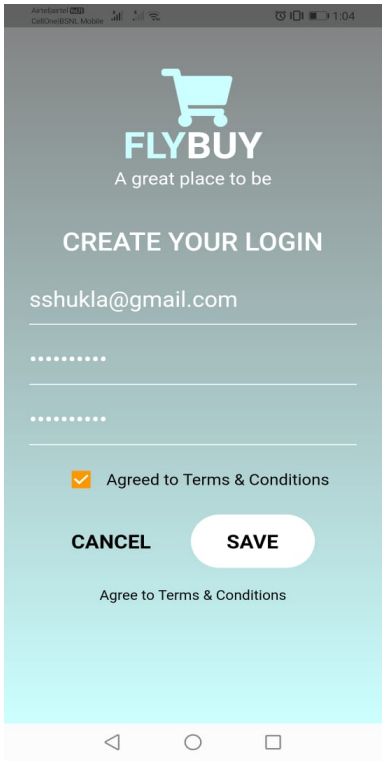

Fig. 10. FlyBuy account creation screen

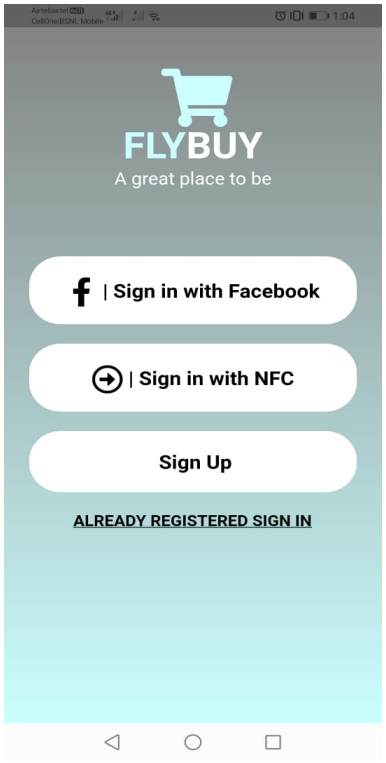

Fig. 11. FlyBuy home page screen with sign in button

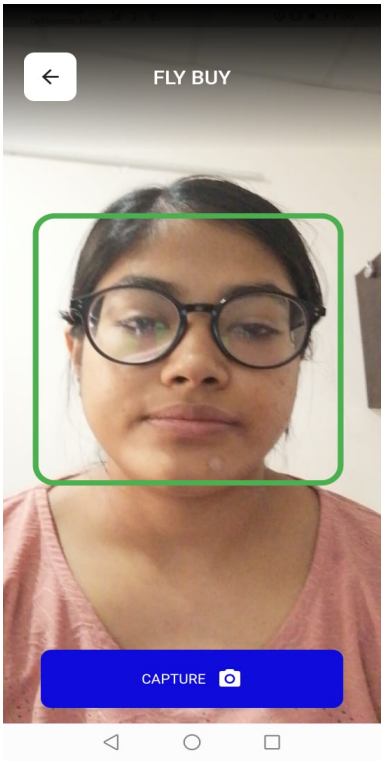

Fig. 12. User taking their face biometric in real time using FlyBuy app

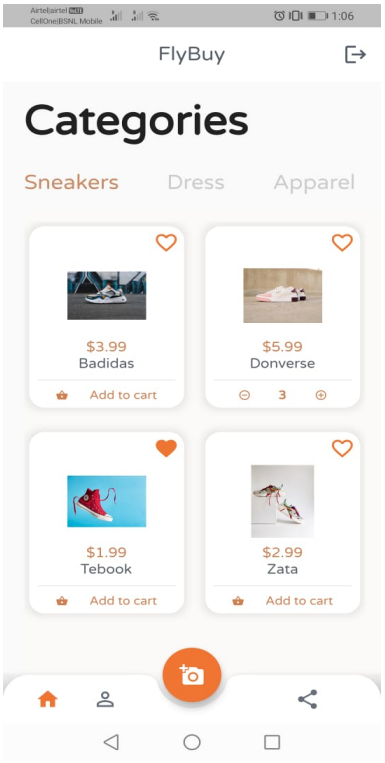

Fig. 13. FlyBuy logged in screen after user authenticates on the second device

TABLE III
COMPARISON IN TERMS OF NUMBER OF TOKENS USED, NUMBER OF TAP AND THEIR SECURITY

Scheme	**Tokens used by scheme**	**Number of touch/tap**	**Token to be remembered by user**	**Additional Needs**	**The need of internet on phone**
3 - U, PWD, OTP on SP	3-U,PWD, SUB	2-U,PWD	Cellphone	N SAASPASS [15]	Google 2 Step [25]
3 - U,PWD, OTP,SUB	2-U,PWD	smartphone	Y Xie et al. [26]	4 - U, PWD, DH, Private Up	3 - U, PWD, OTP on App
2-U,PWD	PC Cam, Smartphone	N Kim et al. [5]	4 - U, PWD, Session ID, Secret Key	3-U,PWD, SUB	4-U,PWD, PC Cam,SUB
Smartphone with GPS	Y Mukho padhyay et al. [27]	3 - U, PWD, Secret key in SP	3-U,PWD, SUB	2-U,PWD	2-U,PWD
Y Dodson et al. [28]	4 - U, PWD, Secret key, QR code	4-U,PWD, QR code, SUB	0 - NIL	Smartphone	Smartphone, TTP
4 - U, PWD, Secret key, OTP CAPTCHA	4-U,PWD, CAPTCHA, SUB	2-U,PWD	NIL	NA Zhu et al. [29]	Y Leung et al. [6]
3-U,PWD CAPTCHA,SUB	2-U,PWD	NIL	NA Tricipher [30]	4 - U, PWD, TPM Secret key, TACS credential	3 - U, SALT, PWD CAPTCHA
2-U,PWD	CAPI driver, Separate hardware, TPM	N RSA SecurID Token [31]	4 - U, PWD, HW token, PIN	4-U,PWD, Secret token, SUB	4-U,PWD, Secret token, SUB
Separate hardware	NA Yubikey U2F [32]	5 - KPUB, KPRIV, Counter, U, PWD	4-U,PWD, Secret Key, SUB	2-U,PWD	2-U,PWD
NA Push login [23]	3 - U, PWD, SP	4-U,PWD, Push notification, SUB	1-U	Smartphone	Separate hardware
3 - U, PWD, master key	1 - SUB	1 - Master PWD	NIL	NA U-PWD	Y Password Managers [33]
3- U,PWD, SUB	2-U,PWD	NIL	NA Proposed scheme	4 - BTADDR1, BTADDR2, PBAPP, Face-Bio	2 - U, PWD
0 - NIL (Face scan)	Smartphone	N			3-Face-Bio, NFC tap, PC tap

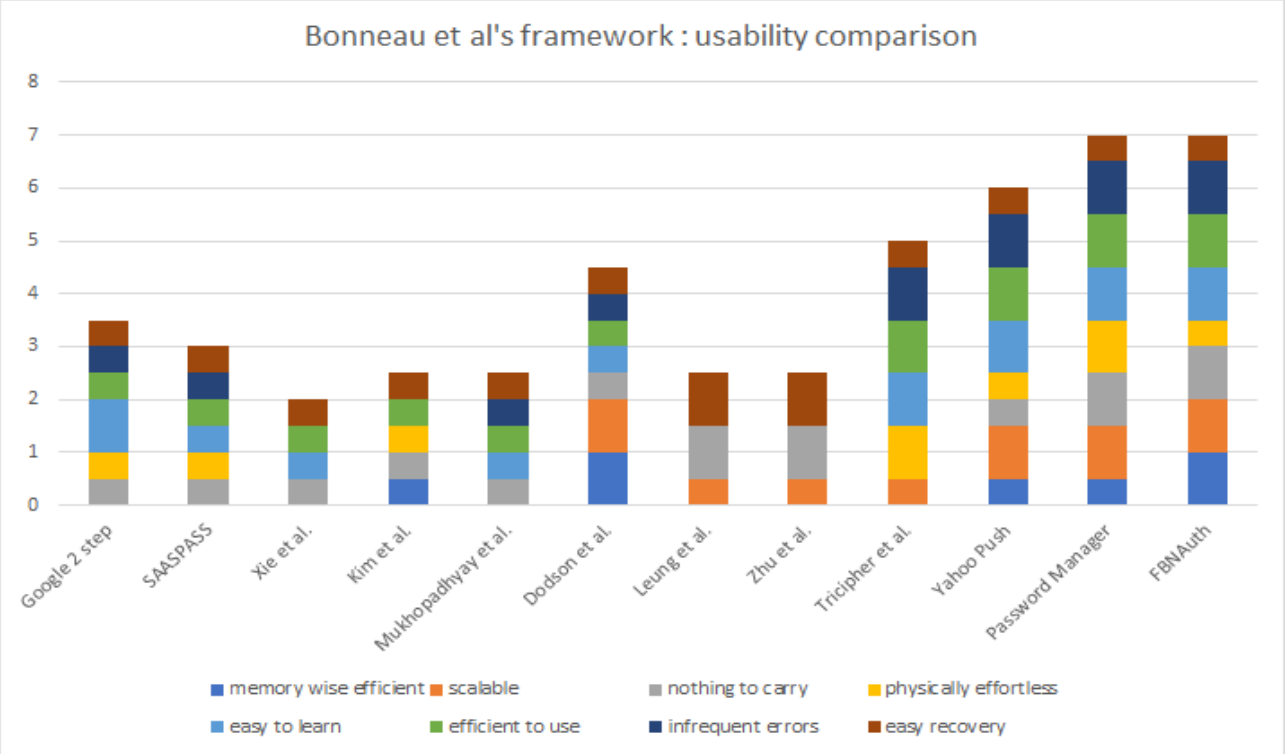

Fig. 14. Usability comparison with existing schemes

model is efficient to use due to two facts: (1) time required for pairing Bluetooth devices (2) tapping using NFC to transfer token takes on an average less than the time required to receive OTP or scanning any Barcode. Apart from this, schemes such as QR code and graphical password requires user understanding for scanning or entering the credentials on the website.

g) **Infrequent error** Several schemes don't offer this benefit. The user might not always obtain a valid SMS or OTP in a stipulated time. Schemes such as QR code / Barcode based schemes have frequent errors since they require the user's to match geolocations based on IP address or perform multiple Barcode scans in some situations. In our proposed scheme, there is very less or no possibility of error, other than the BT address pairing problem, which is rare to occur during the login phase.

h) **Easy recovery** If a user account can be easily recovered even after the loss of a legitimate device or secret token, then it is said to offer this benefit. Several authentication schemes innately provide this easy recovery benefit, but few schemes that use hardware tokens or smartphones might require additional effort for account recovery.

2) **Deployability**

a) **Accessible** The authentication schemes such as moving CAPTCHAs or Barcode/QR code-based schemes use a graphical password or visual password, and hence they are inaccessible to visually or physically impaired users. Such an audience can leverage to touch-based login system. Even if the authentication scheme requires users to gain some technical knowledge before operating their smartphone, then also scheme is considered to be not offering this benefit.

b) **Negligible cost per user** If an authentication scheme requires a separate hardware token or OTP, then it doesn't offer this benefit. However, if it requires having Wifi or Cellular mobile data on a smartphone to complete the authentication process, then it might be considered to be offering this benefit since internet connection is generally available on the smartphone.

c) **Server Compatible** Several authentication schemes are server compatible if they don't require a separate mechanism in order to complete the authentication process, such as generation of certificate or QR code generation or scanning using Android App. In other scenarios where server implementation is platform-dependent or requires specific floating CAPTCHAs appearance, then the scheme doesn't offer server-compatible benefit. Our proposed model and Google 2 step authenticator require minuscule changes at the server end, such as obtaining Bluetooth address or OTP generation. However, traditional username and password-based schemes or password managers completely offer this benefit to the end-user.

d) **Browser compatible** If a user needs to install specific modules or software for their browsers or needs to change their browsers for the authentication scheme to work, then the scheme doesn't offer this benefit. Any scheme providing apps and extensions without the requirement of extra support by browsers such as different versions of scripting language or HTML form related versions, then these schemes are said to offer partial benefit to the user.

e) **Mature** Those schemes that have been rigorously tested and widely adopted by the public are considered to be mature. Those authentication schemes, which are incremental addition to the existing ones, are said to offer partial benefit. Our proposed work also has minuscule verifiable design changes such as obtaining Bluetooth address, capturing real-time face and PBAPP during login from existing MFA schemes.

f) **Non-proprietary** If scheme requires an explicit approval from the developer for their use are said to proprietary and offers no benefit.

3) **Security**

a) **Resilient to physical observation** Any scheme in which all the user credentials cannot be captured by physical observation, such as thermal imaging of keyboard or shoulder surfing or keyboard filming, etc., is said to be resilient to physical observation. Other schemes such as a graphical password or Google 2 step authenticator-based schemes etc., offer this partially as this technique can be used to capture all the user credentials. However, schemes like Push notification based scheme, QR code-based scheme, and our proposed work is secure from all such attacks as credentials cannot be obtained by physical observation.

b) **Target impersonation** Most of the schemes use either QR code or secret key or hardware token or OTP for login purposes which makes it difficult for an attacker with basic knowledge of user details such as age, name, date of birth, etc., to compromise the system and get the user's account access.

c) **Throttled and unthrottled guessing** Only traditional username and password-based schemes are vulnerable to password breaking via throttled or unthrottled guessing because of the presence of QR code or graphical password or hardware tokens, which is not easy to guess by an attacker.

d) **Internal observation** The internal observation can

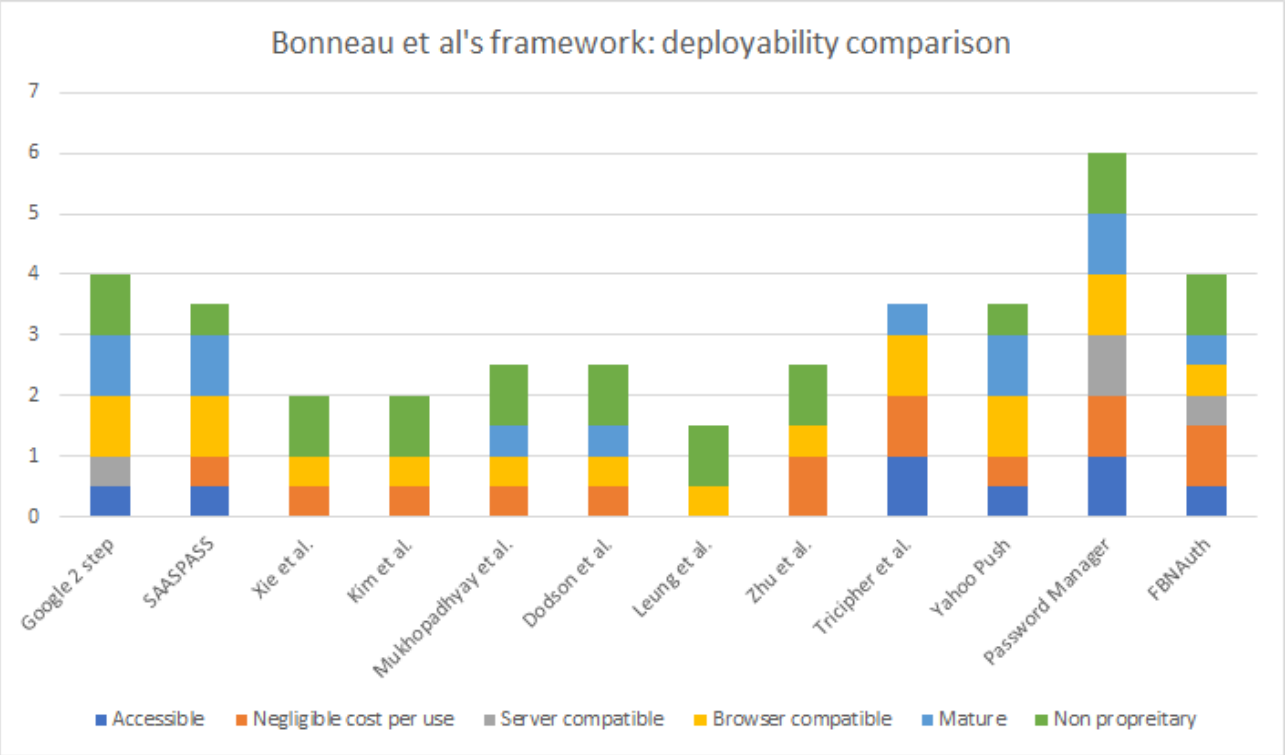

Fig. 15. Deployability comparison with existing schemes

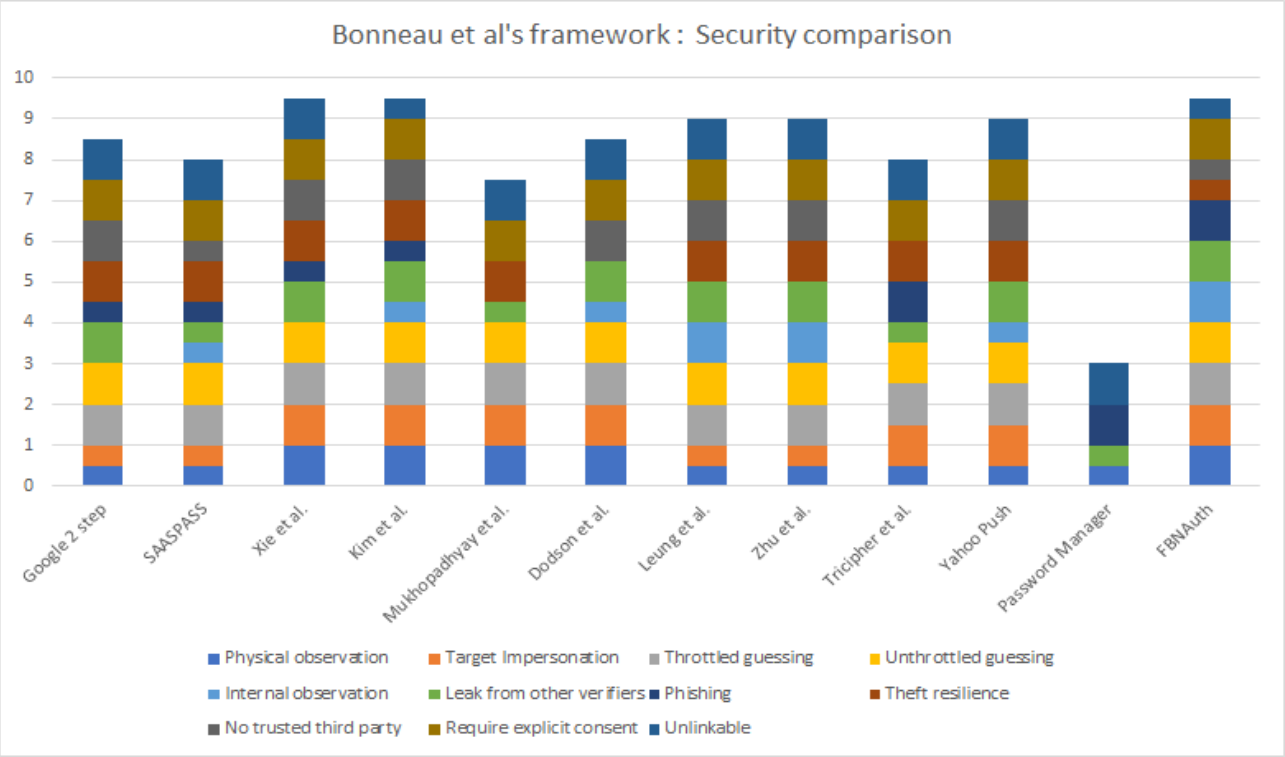

Fig. 16. Security comparison with existing schemes

be carried out by sniffing client and server communication using MBE or by host keylogging. Most of the schemes are unsafe from this attack.

e) **Resilient to leaks from other verifiers** Those schemes that involve a third party during the login phase do not offer this benefit.

f) **Phishing** The authentication schemes which are vulnerable to traditional or RT/CR MITM are also vulnerable to phishing (see Table III)

g) **Resilient to theft** Nearly every authentication scheme is resilient to the theft of smartphone or the hardware token except password manager based schemes and Dodson et al. authentication schemes, in which attacker is capable of directly accessing user account without providing any user information such as PIN on the website if they are in access of the smartphone or PC.

h) **No trusted third party** The authentication scheme that doesn't use a trusted third party during the login process provides this benefit. SAASPASS [35], Mukhopadhyay et al. [27], Tricipher [30] and password manager uses trusted third party for login purpose.

i) **Explicit consent** Nearly every authentication scheme requires the user to explicitly provide their consent during the login phase, except password managers since they used to auto-fill the user information.

j) **Unlinkable** Several schemes offer this benefit and are not used to link with any user. The schemes such as Kim et al.'s [5] is linkable because it uses IP address and this IP address is used to verify the same user identity who is trying to perform login to various websites. Our proposed scheme is somewhat linkable due to the fact that a similar BT device is used to perform login to various websites and hence doesn't offers this benefit.

Our analysis shows that the proposed model performs better in comparison to the other existing schemes, as shown in Figure 9

XI. CONCLUSION AND FUTURE WORK

This paper presents a robust and secure multi-factor passwordless authentication scheme that is capable of handling phishing attacks such as RT MITM, CR MITM phishing, MBE-based attacks, and App spoofing against the above existing schemes [36]–[44] [5], [6], [26], [45] [46], [47], [48], [49], [50], [51], [52], [53] [1], [25], [30], [32], [33], [35], [54]–[65]. The proposed scheme has certain advantages outlined below:

1) The number of credentials that the user needs to remember and enter is reduced to zero. The user doesn't need to remember any token while performing authentication, while other credentials are obtained automatically with the explicit consent of the user, such as capturing face biometric, Bluetooth address (after user enables the Bluetooth functionality on the device), and token transfer using NFC tap (after user enables NFC functionality on the device). This feature ensures security against RT MITM attack since an attacker cannot obtains all the user credentials and relay them in real-time for authentication on a legitimate website.
2) It is not feasible for MBE or any malicious peer App to sniff data or log any user information since the user is never going to provide any credential on the App and also due to same-origin policy, thus avoiding MBE based phishing attacks.
3) The CR MITM and App spoofing attacks were avoided by the use of Bluetooth address and PBAPP.

Another advantage of the proposed model is that it's not client-side dependent, unlike other schemes such as Tricipher. The model was implemented and tested to analyze its efficiency in terms of memory and CPU utilization. The results obtained showed satisfactory performance. Also, a comparison was carried out in terms of usability, deployability, and security against existing schemes, which shows that it performed better than others and ensures the same level of security in comparison with other schemes. In our proposed work, a single Android App was used to log in to the website. This Android App will be provided to the user by a trusted third party (i.e., the organization). Every legitimate website has to fetch, verify and store the Bluetooth address of smartphones of their registered users during the login phase as part of the proposed work. However, our proposed scheme has some current limitations, as discussed below:

- The authentication scheme requires that the user must possess a smartphone when he/she is trying to authenticate to a website using his/her PC. This was reported by a survey [66] that the number of users present across the globe is 6055 billion until 2020, and there are likely chances that this number will increase to 7516 billion by 2026. One of the surveys has mentioned that 4.28 billion people uses internet indicating 90% of the worldwide internet population uses smartphone to go online [67]. Also, two billion NFC-enabled devices like a smartphone are in use today (IHS). In other words, 20%+ of the world's population have access to NFC. This growth will be spurred, in part, by seven billion smartphone subscriptions by 2022 with NFC as a key enabler because it provides consumers the necessary ease of use and convenience making it a useful, universal technology. From the above discussion, it can be assumed that any user using internet services and having technology similar to NFC and BLE enabled smartphones can also use our proposed scheme for authentication purposes.
- The proposed scheme has not been analyzed in terms of security against host-based malware. The malware capabilities include capturing keystrokes of the user, accessing data, resources, and system files, etc. Some malware, such as keystroke loggers, are capable of breaking most of the authentication schemes, such as OTP/PIN-

based schemes. Other malware such as screen loggers is capable of breaking QR code and graphical password-based schemes and even obtain the user credentials sent through separate hardware tokens via memory dumping and analysis. Those malware's who can access the master password or windows password of the user from the system storage can break the Google password manager scheme by compromising user password stored inside Chrome and third-party password managers like LastPass [57] as well. Hence, this attack is not in the current scope of work, can be considered as future work.

REFERENCES

[1] APWG, "Phishing activity trends report, 1st quarter 2021, available:," https://purplesec.us/resources/cyber-security-statistics/ , 8 JUNE, 2021.

[2] "Google. (2015). stronger security for your google account, available:," https://www.google.com/landing/2step/ .

[3] "I. barker. (2015). saaspass makes two-factor authentication available to the masses. available:," https://betanews.com/2015/01/15/sa .

[4] "4 methods to bypass two factor authentication, available:," https://shahmeeramir.com/4-methods-to-bypass-two-factor-authentication-2b0075d9eb5f .

[5] 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*, ser. DIM '13. New York, NY, USA: Association for Computing Machinery, 2013, p. 51–62. [Online]. Available: https://doi.org/10.1145/2517881.2517889

[6] C.-M. Leung, "Depress phishing by captcha with otp," in *2009 3rd International Conference on Anti-counterfeiting, Security, and Identification in Communication*, 2009, pp. 187–192.

[7] "M. m. p. center. (2013). browser extension hijacks facebook profiles. available:," https://blogs.technet.microsoft.com/mmpc/2013/05/10/browserextension-hijacks-facebook-profiles/ .

[8] "C. hoffman. (2017). beginner geek: Everything you need to know about browser extensions. available:," https://www.howtogeek.com/169080/beginner-geek-everything-youneed-to-know-about-browser-extensions/ .

[9] "Mobilefacenets: Efficient cnns for accurate real-time face verification on mobile devices, available:," https://arxiv.org/ftp/arxiv/papers/1804/1804.07573.pdf .

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

[11] B. Parno, C. Kuo, and A. Perrig, "Phoolproof phishing prevention," in *Financial Cryptography and Data Security*, G. Di Crescenzo and A. Rubin, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 1–19.

[12] M. Xie, L. Hao, K. Yoshigoe, and J. Bian, "Camtalk: A bidirectional light communications framework for secure communications on smart-phones," vol. 127, 09 2013, pp. 35–52.

[13] I. Ali, S. Sabir, and Z. Ullah, "Internet of things security, device authentication and access control: A review," 2019.

[14] N. Saxena, J.-E. Ekberg, K. Kostiainen, and N. Asokan, "Secure device pairing based on a visual channel: Design and usability study," *Information Forensics and Security, IEEE Transactions on*, vol. 6, pp. 28 – 38, 04 2011.

[15] "Saaspass 2fa scheme, available:," https://saaspass.com/technologies/proximity-instant-login-two-factor-authentication-beacon/ .

[16] A. Czeskis, M. Dietz, T. Kohno, D. Wallach, and D. Balfanz, "Strengthening user authentication through opportunistic cryptographic identity assertions," in *Proceedings of the 2012 ACM Conference on Computer and Communications Security*, ser. CCS '12. New York, NY, USA: Association for Computing Machinery, 2012, p. 404–414. [Online]. Available: https://doi.org/10.1145/2382196.2382240

[17] "can ios safari access bluetooth device, available:," https://stackoverflow.com/questions/35072438/can-ios-safari-access-bluetooth-device .

[18] "Security key for safer logins with touch in facebook, available:," https://m.facebook.com/nt/screen/?params=%7B%22note$_id$%22%3A10157814544346%2Fnotes%2Fnote%2F$_r$dr.

[19] "Tricipher, "preventing man in the middle phishing attacks with multi-factor authentication," 2016, available:," https://www.helpnetsecurity.com/2005/03/22/tricipher-inc-announces-its-new-authentication-solution-protects-against-man-in-the-middle-phishing-attacks/ .

[20] "Fido alliance, available:," https://fidoalliance.org/ .

[21] "Deconstructing alice and bob, available:," http://www.avispa-project.org/papers/CVB-arspa05.pdf .

[22] L. Malisa, K. Kostiainen, and S. Capkun, "Detecting mobile application spoofing attacks by leveraging user visual similarity perception," in *Proceedings of the Seventh ACM on Conference on Data and Application Security and Privacy*, ser. CODASPY '17. New York, NY, USA: Association for Computing Machinery, 2017, p. 289–300. [Online]. Available: https://doi.org/10.1145/3029806.3029819

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

[24] ""avispa project, available:"," http://www.avispa-project.org/ .

[25] Google, "Google 2 step verification, available:," https://www.google.com/landing/2step/ .

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

[27] Z. Xu, H. Yin, P. Xiong, C. Wan, and Q. Liu, "Short-term responses of picea asperata seedlings of different ages grown in two contrasting forest ecosystems to experimental warming," *Environmental and Experimental Botany*, vol. 77, pp. 1–11, 2012. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0098847211002620

[28] ——, "Short-term responses of picea asperata seedlings of different ages grown in two contrasting forest ecosystems to experimental warming," *Environmental and Experimental Botany*, vol. 77, pp. 1–11, 2012. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0098847211002620

[29] 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.

[30] "Tricipher, "preventing man in the middle phishing attacks with multi-factor authentication", available:," https://www.realwire.com/releases/tricipher-takes-identity-theft-prevention-mobile-with- affordable-portable-man-in-the-middle-protection .

[31] ""rsa securid compromised", available:," https://arstechnica.com/information-technology/2011/06/rsa-finally-comes-clean-securid-is-compromised/ .

[32] "Yubikey, u2f fido standards, available:," https://www.yubico.com/authentication-standards/fido-u2f/ .

[33] "Password manager, available:," https://www.airship.com/resources/explainer/push-notifications-explained/ .

[34] G. Varshney, M. Misra, and P. 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. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2214212618300140

[35] "Saaspass overview, available:," https://betanews.com/ .

[36] C.-Y. Huang, S.-P. Ma, and K.-T. Chen, "Using one-time passwords to prevent password phishing attacks," *Journal of Network and Computer Applications*, vol. 34, no. 4, pp. 1292–1301, 2011, advanced Topics in Cloud Computing. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1084804511000427

[37] V. Mavroeidis and M. Nicho, "Quick response code secure: A cryptographically secure anti-phishing tool for qr code attacks," in *Computer Network Security*, J. Rak, J. Bay, I. Kotenko, L. Popyack, V. Skormin, and K. Szczypiorski, Eds. Cham: Springer International Publishing, 2017, pp. 313–324.

[38] "Product authentication using qr codes: A mobile application to combat counterfeiting download pdf", available:," https://doi.org/10.1007/s11277-016-3374-x .

[39] "Securing sms based one time password technique from man in the middle attack", available:," arXiv:1405.4828 .

[40] J. Xu, J. Qi, and Y. Xi, "Otp bidirectional authentication scheme based on mac address," in *2016 2nd IEEE International Conference on Computer and Communications (ICCC)*, 2016, pp. 1148–1152.

[41] S. K, S. S, and S. M, "Secured mutual authentication between two entities," in *2015 IEEE 9th International Conference on Intelligent Systems and Control (ISCO)*, 2015, pp. 1–5.

[42] Y. Ku, O. Choi, K. Kim, T. Shon, M. Hong, H. Yeh, and J.-H. Kim, "Two-factor authentication system based on extended otp mechanism," *International Journal of Computer Mathematics*, vol. 90, no. 12, pp. 2515–2529, 2013. [Online]. Available: https://doi.org/10.1080/00207160.2012.748901

[43] "The implementation of two-factor web authentication system based on facial recognition", available:," https://doi.org/10.18844/gjcs.v7i2.3448 .

[44] B. Aslam, L. Wu, and C. C. Zou, "Pwdip-hash: A lightweight solution to phishing and pharming attacks," in *2010 Ninth IEEE International Symposium on Network Computing and Applications*, 2010, pp. 198–203.

[45] Y. Lu, L. Li, H. Peng, and Y. Yang, "An energy efficient mutual authentication and key agreement scheme preserving anonymity for wireless sensor networks," *Sensors*, vol. 16, no. 6, 2016. [Online]. Available: https://www.mdpi.com/1424-8220/16/6/837

[46] M. Dhawan and V. Ganapathy, "Analyzing information flow in javascript-based browser extensions," in *2009 Annual Computer Security Applications Conference*, 2009, pp. 382–391.

[47] A. Saini, M. S. Gaur, V. Laxmi, and M. Conti, "Colluding browser extension attack on user privacy and its implication for web browsers," *Computers Security*, vol. 63, pp. 14–28, 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167404816301018

[48] L. V. S. T. C. M. Saini A., Gaur M.S., "Privacy leakage attacks in browsers by colluding extensions." *Information Systems Security. ICISS 2014.*, 2014. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-319-13841-1$_1$5

[49] G. Varshney, M. Misra, and P. Atrey, "Browshing a new way of phishing using a malicious browser extension," in *2017 Innovations in Power and Advanced Computing Technologies (i-PACT)*, 2017, pp. 1–5.

[50] O. S. Discussions, "Dynamically constrained ensemble perturbations – application to tides on the west florida shelf, available:," https://os.copernicus.org/preprints/6/1/2009/osd-6-1-2009.pdf , 1 June, 2009.

[51] N. Fraser, "The usability of picture passwords perturbations – application to tides on the west florida shelf, available:," https://www.tricerion.com/wp-content/uploads/2013/09/Usability-of-picture-passwords.pdf , 1 June, 2009.

[52] R. Dhamija and J. D. Tygar, "The battle against phishing: Dynamic security skins," in *Proceedings of the 2005 Symposium on Usable Privacy and Security*, ser. SOUPS '05. New York, NY, USA: Association for Computing Machinery, 2005, p. 77–88. [Online]. Available: https://doi.org/10.1145/1073001.1073009

[53] G. Varshney, M. Misra, and P. K. Atrey, "Detecting spying and fraud browser extensions: Short paper," in *Proceedings of the 2017 on Multimedia Privacy and Security*, ser. MPS '17. New York, NY, USA: Association for Computing Machinery, 2017, p. 45–52. [Online]. Available: https://doi.org/10.1145/3137616.3137619

[54] APWG, "Phishing activity trends report, 1st quarter 2021, available:," https://docs.apwg.org/reports/apwg$_trends_report_q1_2021.pdf, 8JUNE$, 2021.

[55] Purplesec, "Cyber security statistics 2021, available:," https://docs.apwg.org/reports/apwg$_trends_report_q1_2021.pdf, 4MAY$, 2021.

[56] Owasp, "Qrljacking, available:," https://owasp.org/www-community/attacks/Qrljacking , 4 MAY, 2021.

[57] LastPass, "Lastpass authenitcation, available:," https://www.lastpass.com/ .

[58] "Google account help, available:," https://support.google.com/accounts/answer/7026266?co=GENIE.Platform.

[59] "Biometric advantages and disadvantages, available:," https://www.sestek.com/2016/11/advantages-disadvantages-biometric-authentication/ .

[60] "Fingerprint spoofing, available:," https://fortune.com/2016/02/24/fingerprint-spoofing-easy/ .

[61] "Face recognition facebook, available:," https://www.wired.com/2016/08/hackers-trick-facial-recognition-logins-photos-facebook-thanks-zuck/ .

[62] "Rsa securid, available:," https://www.webopedia.com/definitions/rsa-securid/ .

[63] "Yubikey failed, available:," https://www.yubico.com/blog/ .

[64] "Access control, available:," https://en.wikipedia.org/wiki/Access$_control$.

[65] "Multi-factor authentication, available:," https://searchsecurity.techtarget.com/definition/multifactor-authentication-MFA: :text=Multifactor.

[66] "Number of smartphone users worldwide from 2016 to 2026, available:," https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/ .

[67] "Mobile internet usage worldwide:," https://www.statista.com/topics/779/mobile-internet/: :text=In%202020%2C%20the%20number%20of,mobile%20device%20to%20go%20online. .