A BASTION MOBILEID-BASED AUTHENTICATION TECHNIQUE (BMBAT)

Abdelmunem Abuhasan[1] and Adwan Yasin[2]

[1]Department of Computer Science, Arab American University, Jenin, Palestine
[2]Department of Computer Science, Arab American University, Jenin, Palestine,

ABSTRACT

Despite their proven security breaches, text passwords have been dominating all other methods of human authentication over the web for tens of years, however, the frequent successful attacks that exploit the passwords vulnerable model raises the need to enhance web authentication security. This paper proposes BMBAT; a new authentication technique to replace passwords, that leverages the pervasive user mobile devices, QR codes and the strength of symmetric and asymmetric cryptography. In BMBAT, the user's mobile device acts as a user identity prover and a verifier for the server; it employs a challenge-response model with a dual mode of encryption using AES and RSA keys to mutually authenticate the client to the server and vice-versa. BMBAT combats a set of attack vectors including phishing attacks, man in the middle attacks, eavesdropping and session hijacking. A prototype of BMBAT has been developed and evaluated; the evaluation results show that BMBAT is a feasible and competitive alternative to passwords.

KEYWORDS

Web Authentication, Mobile Authentication, phishing, User Identity, Password.

1. INTRODUCTION

The rapid and continuous advances in web technologies have made the internet industry to be part of almost everyone's today life; from the widely spread social media to companies and financial institutions that offer their services online; thus imposing the very importance of securing the world wide web, especially for end users' private data. One aspect of a website security is the process of authenticating end users to the website services, while the traditional text-passwords are the dominant option for end user authentication, huge efforts -in both the industrial and academic sectors- have been conducted to replace this sticky password scheme; researches concluded that passwords are vulnerable to security attacks [1] including brute-force attacks, guessing, key- loggers, phishing attacks, malwares, eavesdropping, dictionary attacks, etc.

Password authentication have been analyzed over years in an attempt to identify its potential strength and weakness points, a survey of corporate users [2] found that users are confused by the password policy requirements and that they write their passwords on posting notes, thus compromising the security of their accounts. A study on [3] found that users are registered in accounts for which they forgot their passwords and even confused of whether they registered in such accounts or not, so users are usually overwhelmed by the number of passwords they maintain for different websites.

According to [4], three factors contribute to the success of any web authentication scheme: usability, deployability and security. In terms of usability, passwords are considered good in achieving usability measures, as they are easy to learn, efficient to use (just typing a few letters)

and easy to recover from loss as websites usually provide fallback mechanisms for users to recover their passwords.

Passwords most success is in their deployability features; they are accessible, incorporate zero cost for both users and vendors, compatible with both server and browser infrastructure, mature and not proprietary.

Regarding security, passwords evaluate to poor in security measures; they are not resilient to physical observation as they can be automatically observed by key loggers or through a high quality video recording of the keyboard [5]. Also, passwords are vulnerable to dictionary attacks and guessing attacks, not resilient to leaks from other verifiers and most importantly, they are not resilient to phishing attacks.

The poor security evaluation of passwords made users and website vendors to accept to compromise their usability and deployability features for the benefit of more secure authentication schemes. In this paper, we propose and implement an authentication scheme that replaces password authentication, giving the required security level for users and their private data while keeping an accepted level of usability and deployability features, this balance makes the proposed scheme feasible and possible to be implemented and adopted by websites.

This paper is organized as follows, in section 2 we explore the related literature and introduce a set of recent web authentication techniques, section 3 outlines in details the proposed authentication technique including system design, contributions, scenarios of account creation and account sign-in processes, fallback mechanism, system evaluation and security analysis, system implementation and performance analysis. The paper is concluded with future work and conclusions.

1. RELATED WORK

In the literature of human authentication to web applications, many different authentication techniques have been proposed and implemented with different strategies and flavors, including federated sign-on protocols, graphical passwords, dynamic passwords, onetime passwords, QR code based authentication in addition to incorporating other authentication factors in addition to the text-password authentication like biometrics, mobiles, software and hardware tokens. In this section, we review a set of recent authentication schemes.

Federated sign-in protocols allows a web application to authenticate its users using a trusted identity server that verifies the user identity and confirm the user authentication process,while this scheme helps users to stop remembering passwords for the websites they are registered in, it could turn into a complete compromise of all user's accounts in case this authentication to the identity server has been compromised. OpenID [6] and Facebook Connect [7] are two famous examples of this authentication model, Facebook Connect relies on OAuth [8] protocol-,thus giving relying parties access to the users' profiles on Facebook, which is considered an added value for those relying parties to adopt such an authentication scheme, but at the same time making users' private data partially open to service providers.

Ben Dodson et al. [9] proposed Snap2Pass, a mobile based authentication system that aims at replacing the traditional password-based web authentication; leveraging either RSA model or symmetric key encryption. Snap2Pass is based on the challenge-response authentication model; where the server sends a challenge (encrypted token) to the user encapsulated with a QR code, who in turn needs to scan, decrypt and send it back to the server for identity verification. While

this scheme successfully replaces traditional password-based web authentication, two weakness points could harm both the usability and security of this scheme, namely the user's mobile internet connectivity need and the shared key distribution mechanism.

Facebook Account Kit [10] is a recently announced authentication service to enable users to authenticate to their applications using their mobile numbers or email addresses only; the web application service provider needs to integrate with Facebook Account Kit such that users will receive a confirmation SMS code when they wish to login to their web sites.

SQRL [11] is a recent QR code based 2-factor authentication that uses a mobile app with a 256-bit blob of secret data. for each user login attempt, the server generates a QR code encapsulating the server's authentication service concatenated to a long unique randomly generated token, the mobile app then cryptographically hashes the domain name of the site using the user's predefined master key to generate a site specific public and private keys. Then the app signs the entire URL with the site specific private key and sends the signed URL to the server along with the site-specific public key, the server uses the received public key to verify the validity of the signature for the URL and recognize the user using the received public key.

The authors in [12] proposed a user authentication scheme that leverages a user's Android smartphone and SMS to resist password stealing and password reuse attacks, such that the user identity is verified using the mobile application by sending an encrypted one-time secret to the server using SMS such that the server can verify the user's identity.

The authors in [13] proposed the concept of virtual password authentication, where a user-specified function is used to calculate the virtual password with a tradeoff of security for a little more complexity for the user in computing the specified function. Users are authenticated using a dynamic password computed each time using the user's specified function.

Cronto [14] is a commercial transaction authentication system to protect online banking transactions against malware on the user's browser, on this scheme, the user needs to confirm his online transaction using his mobile phone; the website encapsulates an encrypted text containing the transaction details and a onetime code and sends it back in aQR code to the client browser, then the user needs to scan this code into his mobile and decrypt the transaction data per-device key it shares with the bank and display the transaction details in the phone screen, the user then confirms the transaction by entering the transaction password in the browser.

Another authentication scheme proposed by E. Gal´an [15], it relies on generating one-time dynamic and portable URL for each user once he logs in to the web application. This URL is generated specifically for the user in the specified session and then sent to the user through a predefined communication channel (usually SMS or email address).After generating the URL, the server encrypts it using a shared key with the user; when the user receives the encrypted URL he is required to decrypt it and then access the web application through this URL.

Xie et al. proposed CamAuth [16], a 2FA scheme that leverages user's mobile as a second authentication factor, where user identity is proved using a combination of Diffie-Hellman keys exchanged between the client browser (through an extension or Add-on) and the server, 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. Despite the security features that are implemented in CamAuth, the usability and deployability drawbacks could limit the adoption of such an authentication scheme.

Another category of 2FA schemes rely upon client side generation of one time passwords to be used as a second authentication token; a popular 2FA method that falls in this category is Google Authenticator [17]. Google Authenticator (GA) is mobile software that generates offline authentication codes that are used as a second authentication token; such that when the user access his account, he is requested to enter the generated code in addition to his credentials. Dmitrienko et al. [18] performed a security analysis that concluded that such schemes are vulnerable to attacks especially in the registration phase; a PC standing malware can intercept the QR code that encapsulates the pre-shared secrets, then the attacker can initialize his own version of GA and thus being able to generate valid authentication codes for the compromised account.
Czeskis et al. proposed PhoneAuth [19], a 2FA scheme in which the user mobile is considered a second authentication factor in addition to the user credentials; the user is authenticated after signing the login ticket (generated by the server) with the client private key that resides in the user' mobile. The login ticket is communicated back and forth between the client browser and the mobile application through Bluetooth.

2. PROPOSED AUTHENTICATION SCHEME

2.1 Overview

BMBATis designed to meet a set of design guidelines; including achieving mutual authentication between the server and the user through his mobile device, supporting multiple user accounts on different web applications and a secured communication channel (using SSL) between the user mobile and the server.

The proposed authentication method applies the concept of mutual authentication so that the web server proves its identity to the user mobile client and the user mobile client proves its identity to the server before authenticating the user session. As depicted in figure 1,the proposed mutual authentication process incorporates five steps:

Step 1: The user initiates the process by presenting his user name through the web browser to the web server.

Step 2: The web servers validates the user name and prepares the authentication token using the server certificate private key and the shared key, and then encapsulates it with a QR code and send it back to the user browser.

Step 3: The user scans the QR code with his mobile app, processes the login ticket by decrypting it using the server's public key and the shared key to verify the server's identity, and confirms the user identity to the server or extracts the login code and displays it to the user (further discussion of this step is outlined in section 3.3).

Step 4: The application server receives the decrypted nonce, verifies the user identity and authenticates the user's session.

2.2 BMBAT Account Creation

In order for a user to be registered in a website the leverages BMBAT for users' authentication, the user needs to complete a set of steps to identify himself to the website; Figure 2 depicts the main scenario of the registration process. Table 1 summarizes the components notations used in BMBAT.

Step 1: the user is required to specify a valid user name that identifies his account and submit it to the web server along with his mobile IMEI (International Mobile Equipment Identity) code and a valid email address to initiate the account creation process.

Step 2: the web application validates the user name, user email and the IMEI code, and then creates the user account and generates a temporary configuration account credentials to be used by the user to complete the registration process from his mobile device.

Step 3: the user needs to install the BMBAT app in his mobile device if it is not already there.

Step 4: the user uses the temporary configuration account that was generated in step2 to connect to the web app through BMBAT app, at this stage BMBAT completes the user registration by retrieving the server's public key and the server ID and associating it with the user name and then establishes an agreement with the web server on the shared key; using Diffie-Hellman key exchange protocol.

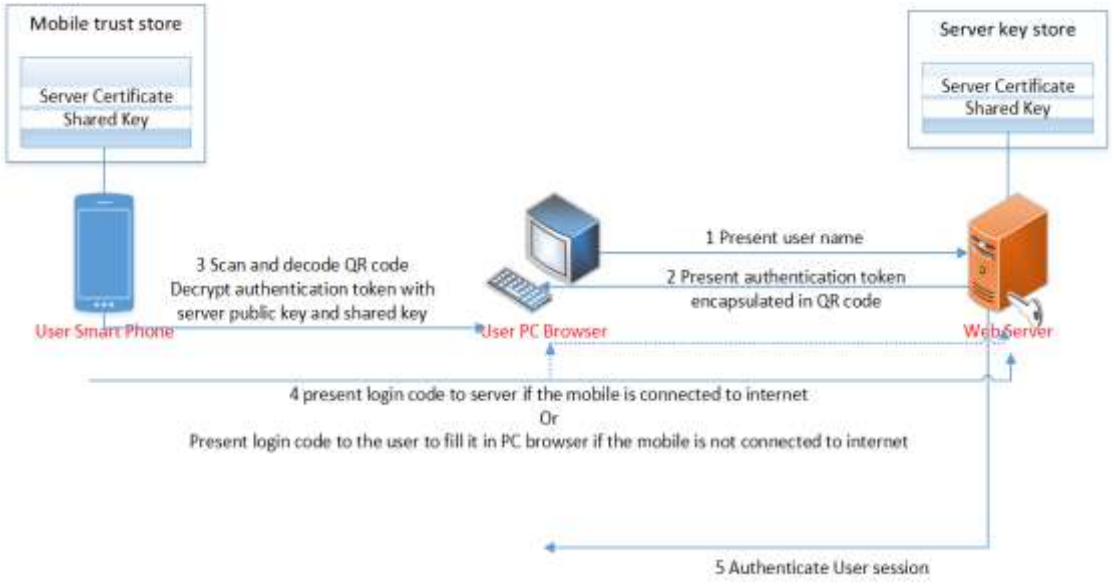

Figure 1: Proposed Mutual Authentication Process

Table 1: Table of notations

Symbol	Name	Description
N	Nonce	Randomly generated 10 alphanumeric characters.
MTS	Mobile Time Stamp	Used to specify a validity time for mobile response.
STS	Server Time Stamp	Used to specify a validity time for the login code
SID	Session Identifier	A web server auto generated string that identifies each web session.
Lcode	Login Code	The challenge that is encoded in the QR code
K	Shared key	256 bit AES key that is shared with the web server for each user.
P_r	Server Private Key	The web server private key (RSA 2048 bit), accessible only to the web server.
P_u	Server Public Key	The web server public key, shared among all users and is publicly available.

Λ	Threshold Value	The validity period (in minutes) of the login code.
ServerID	Server ID	Web application ID to be associated with the user name in BMBAT.
User_ID	User Name	User Name identifier.

The communication between BMBAT and the web application at the registration phase is assumed to be carried over a secured connection (typically HTTPS) to protect the registration data from any unauthorized access.

The association of user name with a server ID is necessary for BMBAT to support multiple accounts on different web applications for the same user. This association is saved in a secure manner in the user's mobile. The shared keys are also stored in a secure trust store in the mobile phone, we think mobile devices trust stores are secure enough to protect the shared key from unauthorized access, thanks to TrustZone technology [20] that is implemented in most modern mobile devices.

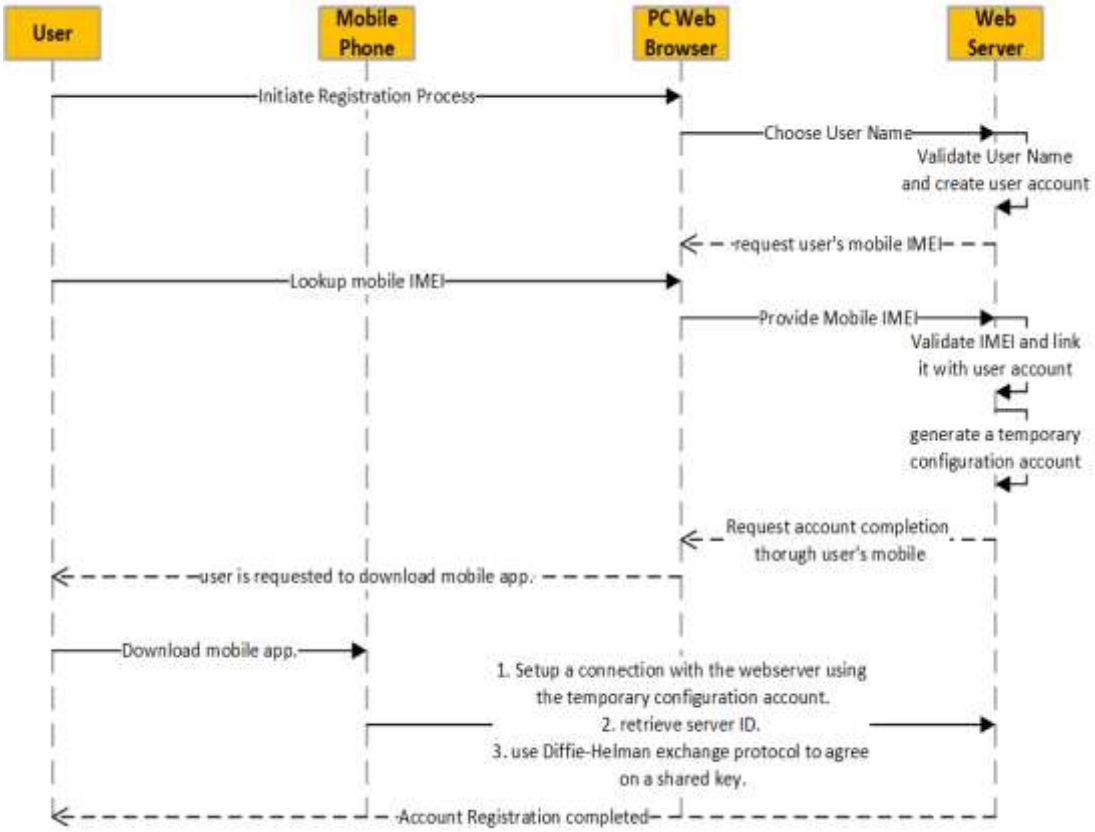

Figure 2: User Registration Sequence Diagram

2.3 BMBATUser Login

The login phase consists of a set of steps as depicted in Figure 3.

Login initiation: when the user wishes to login to the website, he requests the website login page through his "untrusted" browser, then the web server responds with a login page requesting him to fill his user name in addition to filling the Captchacomponent; the requirement of filling the

captcha component is added to prevent attackers from overwhelming the server with auto-generated login requests; built on the assumption that the user name could be publicly known.

User name verification: once the server receives the user name, it verifies that it refers to a registered user, upon success, the server initiates the task of creating the user authentication token.

User Authentication Token: in response to the user login request, the web server prepares a Login Code for that user's session; it generates a random session nonce (N) and associates it with the user session (SID), server current timestamp (STS) and user name, then it encrypts it with the shared key (K), the resulting cipher is then concatenated with the user name and Server Authentication Service API and then digitally signed with the server's private key (P_r). the resulting authentication token and ServerID are encoded in a QR code and sent back to the user's browser. An XML representation of the authentication token contents could be as follows:

{
ServerID: *5T208*
Token: *"15g8T45Krco…"*
Server_Auth_API: *"https://login.examplesite.com"*
}.

The login code generation algorithm is depicted in figure 4.

Response: The encoded QR code is sent back as a response to the user's browser, the response

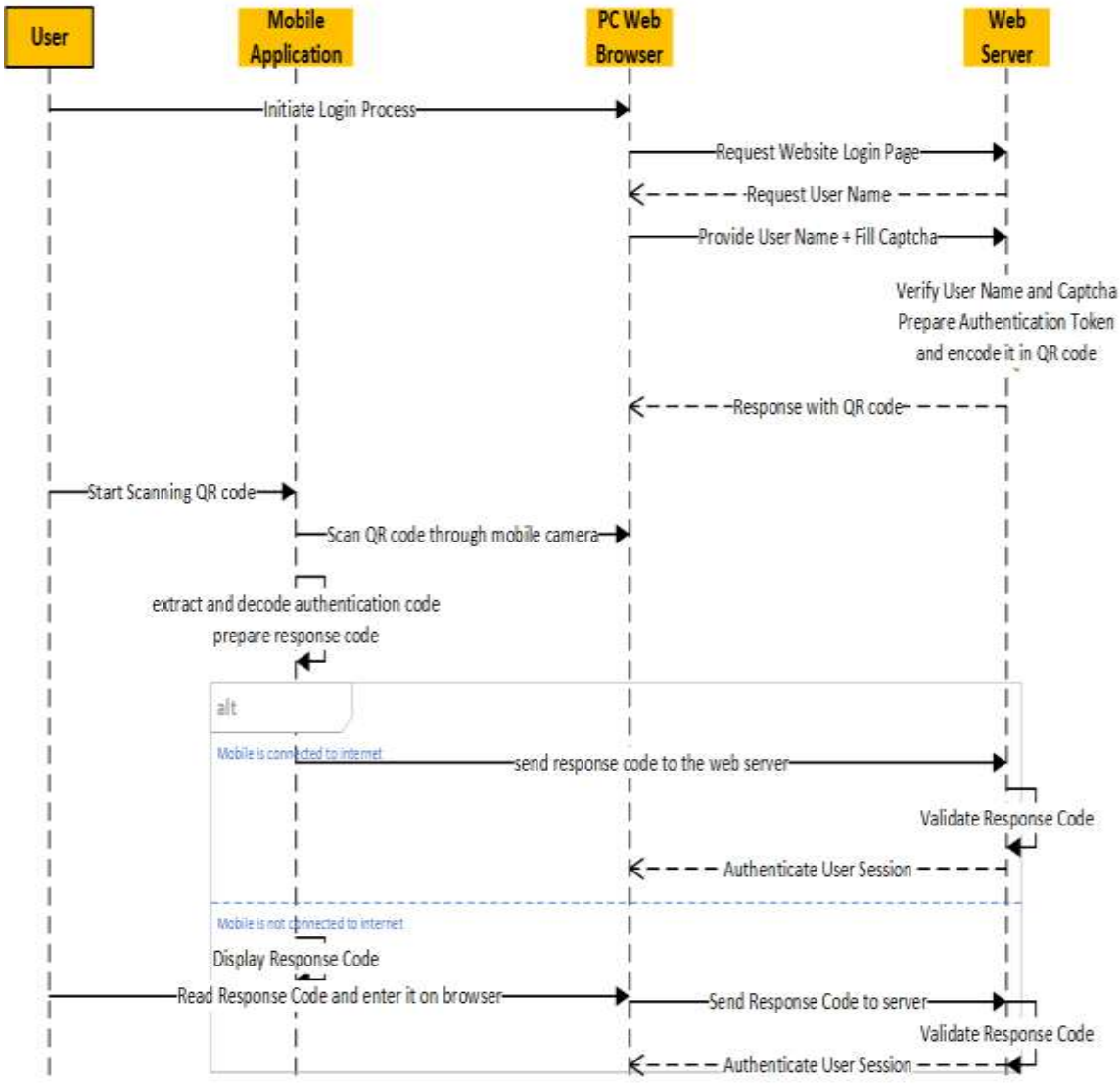

Fig. 3. Login Phase Sequence Diagram

page also offers an option for the user to enter the deciphered login code as will be explained next.

QR code scanning: now the user needs to scan the QR code that is displayed on the PC browser to complete authenticating himself to the web site, this scan is done using the user's mobile application that was associated with his account in the registration process.

```
//this algorithm generates the authentication token
Begin
User_id=getParameter(User_ID);
SID=getSessionIdentifier();
N=generateRandomNonce();
STS=getServerTimeStamp();
K=lookupSharedKey(User_ID);
Token=concatenate(N,SID,User_ID,STS);
Token=encrypt(K,Token);
Pr=lookupServerPrivateKey();
encryptedToken=encrypt (Pr, Token);
response=concatenate(ServerID,Server_Auth_API,
encryptedToken);
saveAuthenticationInfo(SID,N,STS,User_ID);
sendResponseQRToClient(response);
End;
```

Fig.4. Token Generation Algorithm

```
//this algorithm prepares the authentication token using t
//server's private key and the shared key to be used in the
//mutual authentication process
Begin
authToken=scanQRCode();
ServerID=extractServerID(authToken);
Token=extractToken(authToken);
Pu=lookupServerPublicKey(ServerID);
decryptedToken=decrypt (Pu, Token);
Server_Auth_API= extractServerAuthApi(authToken);
User_ID=extractUserID(decryptedToken);
encryptedNonce=extractNonce(decryptedToken);
K=lookupSharedKey(User_ID,ServerID);
decryptedToken=decrypt(K, encryptedNonce);
SID=extractSID(decryptedToken);
N=extract(decryptedToken);
STS=extractSTS(decryptedToken);
If validSTS(STS,λ) then
  If mobile_connected_internet then
    Response=encrypt(K,N);
    SendResponse(Response+User_ID+SID+MTS,
Server_Auth_API);
  Else
    displayToMobileScreen(N);
  End if;
End if;
End;
```

Fig. 5. QR code processing algorithm

```
// this algorithm applies the server steps to validate
//the client authentication
Begin
Token=getParameter(Token);
User_ID=extractUserID(Token);
SID=extractSID(Token);
MTS=extractMTS(Token);
encryptedNonce=extractEncryptedNonce(Token);
if validSID(SID, User_ID) then
  K=lookupSharedKey(User_ID);
  N=decrypt(K, encryptedNonce);
  If N==loadSavedNonce() and validMTS(MTS,λ)
then
    Authenticate_user_session();
  Else
  Deny_user_session();
  End if;
end if;
End;
```

Fig. 6. Server authentication algorithm

Once the authentication token is extracted from the QR code in the mobile application, the following actions and options take place (see figure 5):

- The authentication token is decrypted using the server's public key, to extract the user name and the encrypted nonce.

- BMBAT looks up the shared key based on the user name and the ServerID.
- The encrypted nonce is then decrypted using the pre-shared key to extract the login token and the associated time stamp (STS) and session identifier SID.

- The login token is now available for BMBAT, and its validity is determined based on the time stamp of the token creation time and a threshold value (λ).

- The next action is determined based on whether the mobile is connected to the internet or not; If the mobile has access to the internet, the login token is augmented with the mobile timestamp and SID and sent back to the server directly (Server_Auth_API) over a secure connection. The server verifies the received login token against the user's session and automatically authenticates the user session if the verification succeeds (see figure 6). In case the mobile is not connected to the internet, the mobile application extracts the nonce N from the login code and displays it to the user. Then the user is required to enter this nonce manually in his browser. The server verifies the received nonce against the user's session and authenticates the user session if the verification succeeds.

2.4 Fall-back Mechanism

In case the user mobile is no longer available (due to theft, damage, etc..), the user needs to disassociate his account from the not accessible mobile device, BMBAT offers the option for the user to deactivate BMBAT authentication by visiting all registered websites and requesting sending a deactivation email to his email address, which includes a link through which he can remove his mobile device from the trusted devices.

Also another mitigating factors could be used to protect the user accounts in case the mobile is lost; it is possible to damage the mobile phone data using the privileges granted by the phone

vendor, also the user is encouraged to configure his mobile so that a pass code (or maybe a finger print) is needed before opening BMBAT app, in this case the attacker needs to bypass the passcode before being able to impersonate the user.

2.5 BMBATEvaluation and Security Analysis

In order to evaluate BMBAT usability, deployability and security features, we benchmarked it to another four known authentication schemes; based on the web authentication assessment framework proposed by Bonneau et al. [1]. A set of 25 measures have been applied to BMBATand compared to other four popular authentication schemes; Passwords, Google 2 step-verification [17], PhoneAuth [19] and CamAuth [16], the comparison results are shown in Table 2.
The comparative evaluation results show that BMBAT is highly competitive to existing authentication techniques. In terms of usability, BMBAT offers the benefit of eliminating the need for users to remember their passwords and neutralizes password entry errors by eliminating the use of passwords at all, while offering somewhat the benefit of "Nothing to Carry"; as we consider the mobile device to be always side by side to the user.

In terms of deployability, BMBAT is an accessible and zero-cost for the user, but we could not assume at a mature authentication technique as this property needs to be test thoroughly in production environments. The security features of BMBAT enable it to highly compete with similar authentication techniques, the offered security features are discussed as follows:

1. Resilient to Physical Observation: BMBAT achieves this property due to the fact that it does not rely on password authentication, so any physical observation of the authentication process will never reveal any clue on the authentication details. In case the user needs to enter the login code in his browser, the observation of this code is useless as it is a session-based onetime code.

2. Resilient to Targeted Impersonation: an attacker who possess a knowledge of the user's personal details (such as birthdate, relative names etc.) will be not able to impersonate the user, as none of such information is needed or related to any step in BMBAT authentication.

3. Resilient to Throttled and Unthrottled Guessing: an attacker's chance to succeed to guess or brute-force the shared key between the user and the server is very low, and trying to brute-force the authentication nonce is not useful for an attacker as it is valid only for the current user session.

4. Resilient to Internal Observation: An attacker cannot impersonate a user by intercepting the user's input from inside the user's device (e.g., by keylogging malware) or eavesdropping on the clear text messages communicated between prover and verifier; BMBAT achieves this property by using a session-based onetime token that renders its usage again of no benefit, in addition all data communication is assumed to be carried over a secure channel.

5. Resilient to Leaks from Other Verifiers: BMBAT authentication parameters (shared keys) are dedicated for users in a specific web site, so that a successful dictionary attack on a web site or the user mobile (and thus compromising shared keys) will never make it possible to compromise user accounts in other different websites.

6. Resilient to Phishing: BMBAT neutralizes phishing attacks by eliminating their core attack principle; i.e. no password to be stolen; The phone sends the shared secret, and will only

send it to the web site associated with the user name and ServerID that were extracted from encrypted QR code login ticket.

7. Resilient to Theft: we rated BMBAT to be resilient to theft as there are a set of solutions to prevent a stolen user's mobile
from being used to access the linked accounts; starting from locking codes that are available on modern mobile phone, to the possibility of deactivating the phone online by service providers. Nevertheless, BMBAT will offer the option for a user to deactivate his account association with the mobile phone in case it was lost; the user can access his account online using a variety of ways including master password or onetime passwords.

8. No Trusted Third Party: BMBAT does not rely on a trusted third party (other than the user mobile and the server) who could, upon being attacked or otherwise becoming untrustworthy, compromise the prover's security or privacy.

9. Requiring Explicit Consent: BMBAT authentication will never be initiated without an explicit action from the user to request authentication; this is achieved through requesting a user name and a Captcha code and then scanning the QR code by the user's mobile.

10. Unlinkable: this property "measures whether Colluding verifiers can determine -from the authenticator alone, whether the same user is authenticating to both" [4]; we rate BMBAT as Unlinkable as the parameters that identify a user (shared key and user name) need not to have any thing in common when the user is registered to more than one website.

Table 2. Comparison of BMBAT, Passwords, Google 2-Step Verification (2SV), PhoneAuth (in strict mode) and CamAuth.Y=offers the benefit, S=somewhat offers the benefit.

		Passwords	Google 2SV	PhoneAuth	CamAuth	BMBAT
Usability Features	Memorywise Effortless					Y
	Scalable for Users	Y				S
	Nothing to Carry	Y	Y	S	S	S
	Quasi Nothing to-Carry	Y	Y	Y	Y	Y
	Easy to Learn	Y	S	Y	S	Y
	Efficient to Use	S	S	S	S	Y
	Infrequent Errors	Y	S	S	S	Y
	Easy Recovery from Loss	Y	S	S	S	S
Deployability Features	Accessible	Y	S	Y	S	Y
	Negligible Cost Per User	Y		S	S	Y
	Server Compatible	Y		S	S	Y
	Browser Compatible	Y	Y	S	S	Y
	Mature	Y	Y			S

	Non Proprietary	Y		Y	Y	Y
Security Features	Resilient to Physical Observation			Y	Y	Y
	Resilient to Targeted Impersonation	S	S	Y	Y	Y
	Resilient to Throttled Guessing		Y	Y	Y	Y
	Resilient to Unthrottled Guessing			Y	Y	Y
	Resilient to Internal Observation			S	S	Y
	Resilient to Leaks from Other Verifiers		Y	Y	Y	Y
	Resilient to Phishing		Y	Y	Y	Y
	Resilient to Theft	Y	Y	Y	Y	Y
	No Trusted Third Party	Y	Y	Y	Y	Y
	Requiring Explicit Consent	Y	Y	Y	Y	Y
	Unlinkable	Y	Y	S	Y	Y

2.6 **BMBAT Contributions**

BMBAT is designed to offer a better security than text passwords with minimum compromise of usability and deployability for both end users and service providers, it protects against the following attack vectors:

1. Phishing attacks: while traditional phishing attacks succeed to compromise the user's credentials on password-based authentication schemes; the proposed authentication scheme neutralizes them completely; as there is no password for the user to be compromised, the user is identified through his mobile device.

2. Man in the middle attacks:In this class of attacks, the attackers situate themselves between the user and the original web site, and proxy all communications between the user and the real web site, from this point, the attacker can observe and record all transactions including the user's credentials. For such an attacks to succeed, the attacker must be able to direct the user to the attacker's proxy server instead of the real server. This may be accomplished using a number of methods including Transparent Proxies, DNS Cache Poisoning, URL Obfuscation and Browser Proxy Configuration. The proposed scheme combats this type of attacks by:

    - Mutually authentication both parties- the user and server- with the pre-shared key and server's private key respectively.

    - The authentication in the proposed system relies upon a session-based login code; which renders it useless for any attacker to reuse another session's login code.

3. eavesdropping: the proposed system assumes it is possible for an attacker to passively or actively eavesdrop on any network communication between the user and the server in the login phase; as the traffic data is either protected by a dual encryption mode with the pre-shared key and server's private key or the data is session based; i.e. it is valid only for the current session.

4. Dictionary attacks: the proposed system is not immune to dictionary attacks; as the keys shared with the system users need to be maintained in somehow on the server machine. Any compromise of the shared keys would put the user accounts in risk.

5. Session Hijacking: BMBAT assumes that the data communicated between different parties in the authentication process is protected using HTTPS protocol, such that it is not possible for an attacker to exploit a valid session to gain unauthorized access to the website in behalf of the user.

2.7 BMBAT Implementation and performance analysis

We have implemented a prototype system for BMBAT authentication protocol, including a web application and mobile application to act as an identity prover. We built a java-based web application that handles a set of server side services for the authentication protocol including token generation, encryption, QR code generator and an authenticator. the client's mobile application is developed on android 5.1, and its compatible with android 2.2 and upward platforms, the mobile application is responsible for device registration, maintaining user shared keys, confirming and completing the authentication process, the application uses the necessary APIs for key exchange, storage, encryption, decryption and communications with the server.
To assess the performance of BMBAT, we conducted a performance test using an emulator in android studio with the following specifications:

- Device: Nexus 5
- CPU: x86
- RAM: 1.5 GB
- Platform Version: Android 5.1 (lollipop)
-

In this evaluation we measured two metrics of the non-functional requirements of the system:

- Response time: the time in seconds that BMBAT mobile app takes to execute the algorithm of processing the QR code figure 5.
- Memory usage: the volume in megabytes that BMBAT decryption process used in the RAM of the mobile execution environment.

The results of the performance test showed that BMBAT -in average- spent 4.3 Milliseconds in executing the algorithm of processing the login code, while consuming 0.08 MB of memory. The whole memory reserved by the application was 2.57 MB. Table 3 depicts the test results of six runs of the algorithm of decrypting the login code using the emulator.

Table 3: login code processing algorithm performance on the emulator

Run #	Execution Time (ms)	CPU Usage	Memory Usage (MB)
1	7	15%	0.09
2	4	12%	0.08
3	6	12.5%	0.07
4	5	9.5%	0.06
5	8	16%	0.07
6	4	18%	0.04

The complete authentication process will include also the time spent in scanning the QR code and communicating the login token to the server, the scan process is assumed intuitive and easy operation for most users, and the process of sending back the login token to the server is an automated process that is expected to add a very little time fraction (in milliseconds).

These performance test results indicate that BMBAT implementation in most modern mobile devices will be feasible and the response time will be accepted by users, making it possible to adopt such an authentication scheme at a compromise of a couple of seconds latency.

3. FUTURE WORK

The emergence of the Internet of Things "IoT" opens the door for smaller and maybe more ubiquitous user devices that could be a better replacement for user's mobile phone for identity proof. So the future may raise the need to reevaluate the cost of executing cryptographic algorithms (either symmetric or asymmetric) on such devices. One prominent choice for replacing such algorithms in BMBAT is to use the less costly elliptic curve cryptography which requires smaller keys compared to AES or RSA but offers the same security levels, thus needs far less processing than current RSA or symmetric key cryptography.

4. CONCLUSION

In this work we presented BMBAT, a new web authentication scheme that employs the user's mobile phone as an identity proofer. This technique employs a dual encryption mode (RSA and AES) to implement a challenge –Response mechanism that enables the web server to identify the user identity in an easy and smart manner. The mobile application completes the response phase by either sending the response directly to the server or by displaying the response code to the user in case the mobile is not connected to the internet. Our evaluation and security analysis of BMBAT concluded that it is a competitive alternative to traditional password-based web authentication and that it overcomes all security breaches that are plausible to passwords. Moreover, we implemented a prototype of BMBAT to prove that it is applicable and a feasible alternative to current web authentication methods.

REFERENCES

[1] 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," University of Cambridge Computer Laboratory, Tech Report 817, 2012, www.cl.cam.ac.uk/techreports/UCAM-CL-TR-817.html.

[2] A. Adams and M. Sasse, "Users Are Not The Enemy," Commun. ACM, vol. 42, no. 12, pp. 41–46, 1999.

[3] S. Gaw and E. W. Felten, "Password Management Strategies for Online Accounts," in ACM SOUPS 2006: Proc. 2nd Symp. on Usable Privacy and Security, pp. 44–55.

[4] J. Bonneau, C. Herley, P. C. v. Oorschot, and F. Stajano, "The quest to replace passwords: A framework for comparative evaluation of web authentication schemes," in Proceedings of the 2012 IEEE Symposium on Security and Privacy, 2012, pp. 553–567.

[5] D. Balzarotti, M. Cova, and G. Vigna, "ClearShot: Eavesdropping on Keyboard Input from Video," in IEEE Symp. Security and Privacy, 2008, pp. 170–183.

[6] D. Recordon and D. Reed, "OpenID 2.0: a platform for usercentric identity management," in DIM '06: Proc. 2nd ACM Workshop on Digital Identity Management, 2006, pp. 11–16.

[7] Facebook Connect,https://developers.facebook.com/docs/facebook-login/overview, accessed November 2016

[8] D. Recordon and D. Hardt, "The OAuth 2.0 Protocol," April 2010, tools.ietf.org/html/draft-hammer-oauth2-00.

[9] Ben Dodson, Debangsu Sengupta, Dan Boneh, and Monica S. Lam, "Secure, Consumer-Friendly Web Authentication and Payments with a Phone".
[10] Facebook Account Kit, https://developers.facebook.com/products/account-kit, accessed November 2016
[11] Secure Quick Reliable Login, https://www.grc.com/sqrl/sqrl.htm, accessed November 2016
[12] Prajitha M V, Rekha P, Amrutha George A, "A Secured Authentication Protocol Which Resist Password Reuse Attack ", IEEE Sponsored 2nd International Conference on Innovations in Information Embedded and Communication Systems ICIIECS'15, , 2015
[13] P. Umadevi and V.Saranya, "Stronger Authentication for Password using Virtual Password and Secret Little Functions", ICICES2014 - S.A.Engineering College, Chennai, Tamil Nadu, India, IEEE 2014.
[14] Cronto, www.cronto.com/, accessed Novemeber 2016
[15] E. Gal´an and J.C. Hern´andez–Castro and A. Alcaide and A. Ribagorda ,A Strong Authentication Protocol based on Portable One–Time Dynamic URLs,,2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology.
[16] Mengjun Xie, Yanyan Li, Kenji Yoshigoe, Remzi Seker, Jiang Bian, "CamAuth: Securing Web Authentication with Camera", 2015 IEEE 16th International Symposium on High Assurance Systems Engineering.
[17] Google 2step verification,http://www.google.com/landing/2step/, accessed November 2016
[18] Alexandra Dmitrienko, Christopher Liebchen, Christian Rossow, Ahmad-Reza Sadeghi, "Security Analysis of mobile two-factor authentication schemes", Intel technology journal volume 18, Issue 4, 2014.
[19] 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, 2012, pp. 404–414.
[20] TrustZone technology,http://www.arm.com/products/processors/technologies/trustzone/, accessed November 2016
[21] Rescorla, E., Diffie-Hellman Key Agreement Method, RFC 2631, IETF Network Working Group, http://www.ietf.org/rfc/rfc2631.txt

AUTHORS

Abdelmunem Abuhasan is a Master student at the Arab American University with particular interests in computer security, web security and software engineering. He is working since ten years as the manager of software development department at the Arab American University. He holds a B.A. in Computer Science from the Arab American University.

Adwan Yasin is an associate Professor, Former dean of Faculty of Engineering and Information Technology of the Arab American University of Jenin, Palestine. Previously he worked at Philadelphia and Zarka Private University, Jordan. He received his PhD degree from the National Technical University of Ukraine in 1996. His research interests include Computer Networks, Computer Architecture, Cryptography and Networks Security.