

# 3LS-Authenticate: an e-Commerce Challenge-Response Mobile Application

Rania. A. Molla

Department of Computer Science  
Collage of Computing and Information Technology  
King Abdulaziz University  
Jeddah, Kingdom of Saudi Arabia  
[rmolla@kau.edu.sa](mailto:rmolla@kau.edu.sa)

Imed Romdhani, Bill Buchanan

School of Computing  
Edinburgh Napier University  
Edinburgh, UK  
[I.Romdhani@napier.ac.uk](mailto:I.Romdhani@napier.ac.uk)  
[B.Buchana@napier.ac.uk](mailto:B.Buchana@napier.ac.uk)

**Abstract** - The rapid growth of e-commerce has been associated with a number of security concerns, which challenge its continual success. In view of this, an investigative study determining the most secure and convenient solution to protect online clients has been conducted. It was found that employing mobile phones to authenticate clients, through Out-Of-Band (OOB) communication channels, was the best solution to overcome security threats, such as Man-In-The-Browser (MITB) attacks. Therefore, a simple, yet highly secure, mobile application was developed to authenticate online clients within e-commerce applications using QR code capturing.

This paper introduces the “3LS-Authenticate” mobile-application, which captures an encrypted QR code from a server’s web-browser, and performs three levels of security to authenticate clients. It also presents results of verification of the proposed protocol, using the Scyther security protocol verification tool.

**Keywords** - Authentication; Man-In-The-Browser (MITB) attack; QR code.

## I. INTRODUCTION

With the rapid evolution of the smart phone industry, and the increase of online fraud and web-browser-based security vulnerabilities, there has been a developing trend towards using mobile phones as an authentication system in e-commerce. These solutions are used for online identity authentication [1] [2] [3] [4] [5], secure virtual private network login [6] [7] [8], and online financial transactions [5] [9] [10] [11] [12] [13].

This paper introduces a new mobile phone application, termed “3LS-Authenticate”, which provides a secure system for performing a three-level security authentication process for clients within the context of e-commerce applications.

The proposed application is a part of a larger system termed the “Mobile User Authentication system” or (MUAS) [14]. The MUAS has been designed to transition from an insecure network (the Internet) to an OOB, in order to avoid MITB attack. Therefore, the MUAS process is split into two protocols: challenge-generation, and response-generation. This separation is performed due to the QR code capturing process, which is responsible for initiating the response-generation

protocol with a new connection to the server. Fig. 1 illustrates a high-level representation of the MUAS.

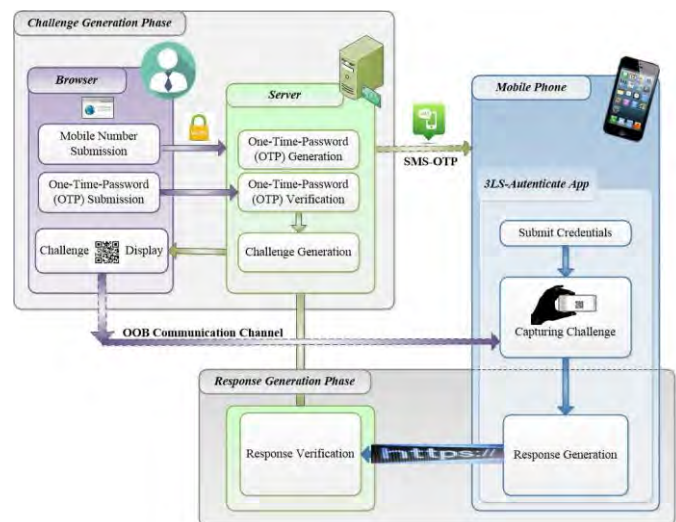


Figure 1. MUAS high-level representation

Challenge-generation represents the first phase of the MUAS protocol. This generates the QR code and displays it on the web-browser. The protocol takes place within a secure SSL/TLS communication channel between the browser (client) and the merchant (server), in order to mutually authenticate both parties, and to submit the required information to the server in a secure manner. The protocol employs one-time password (OTP) authentication to ensure that the connecting party (browser) is the legitimate holder of the authentication device (mobile phone). The server then verifies the received OTP to form the challenge-encrypted QR code, and sends it through a secure channel to be displayed within the browser, in order to be captured by the mobile phone. Next, the response-generation protocol continues the MUAS process, but in a different communication channel. The response-generation protocol takes place between the server (merchant) and the client’s mobile phone (3LS-Authenticate) within a secure SSL/TLS communication channel.

In developing the “3LS-Authenticate” mobile application, as a part of an e-commerce authentication system that overcomes MITB attacks, the following questions arise:

- Who will conduct the authentication process?
- What kind of information must be provided by the client?
- Can the system ensure mobile-phone-holder legitimacy?
- If so, how do we verify the identity of the client and merchant via QR code?
- How can the system overcome the risks of MITB attacks?
- What kind of technologies and protocols were used to perform mutual authentication between participants, and to avoid malicious users and replay attacks?
- Does the system satisfy e-commerce security requirements?
- Is the system adequately secure against attacks?

“3LS-Authenticate” is implemented using Objective-C on Xcode 6.1.1, with most of the application’s functions being performed using software components and libraries provided by Chilkat Software Inc. [15]. The client’s digital certificate (.pfx) is installed on the mobile phone, for use in the cryptography step within the authentication process.

The remainder of this paper is divided into seven sections: II) lists research contributions; III) presents the application’s architecture; IV) describes the application’s core algorithm; V) deals with response-generation protocol design; VI) presents the security analysis, along with protocol evaluation criteria and verification results, using Scyther; VII) discusses related work; and VIII) presents the conclusion.

## II. RESEARCH CONTRIBUTION

This study describes the MUAS secure authentication system, which is designed to efficiently authenticate clients within e-commerce websites in a secure manner, using smartphones. MUAS authenticates clients using a two-factor authentication method within the context of challenge and response generation protocols. Both protocols are built on top of the SSL/TLS protocols in order to secure all connections and communications between participants. These protocols also employ cryptographic nonce to prevent replay attacks. Thus, MUAS ensures authentication and secrecy. The main contributions of this study can be summarized as follows:

- **Designing a user-friendly and non-confidential input-based authentication system.** The proposed MUAS does not require registration with e-commerce websites. The client is required to submit only minimal, non-confidential information. This step bypasses the traditional password-based authentication mechanisms, instead relying upon a more secure mechanism (OTP). The MUAS approach can be considered an economic authentication system within the context of a server’s storage media, as no registration information is saved for clients, thus saving server storage. Instead, a record for

each real-time client is created and saved within a server-specific table.

- **Overcome MITB attack by designing a two-factor mobile phone-based authentication system.** Both MUAS authentication factors are based on proof-by-possession. The first factor transitions the mechanism from an insecure network to a direct OOB communication channel (SMS OTP-based channel), in order to ensure that the submitted mobile phone number is actually owned by the client initiating the authentication process. As the system is still prone to MITB attack (because a user submits the OTP through the browser), a second authentication mechanism is needed in order to minimize browser utilization. This second factor overcomes MITB attack by transitioning from an insecure network to an indirect OOB communication channel (QR code).
- **Generating unique and useful cryptographic-based QR code contents.** For each session, a one-time secure cryptography-based QR code is generated. Its contents include the server’s cryptographic nonce and URL. The nonce, generated by the server, is never repeated during a session. Thus, it is unique. In a scalable system, a nonce is used to link the received response from the mobile phone to that client who initiated the authentication process using the browser. The server’s URL is exploited to establish a secure SSL/TLS connection between the mobile phone and the server, in order to send generated cryptography-based responses for verification.
- **Designing a three-level secure mobile phone application.** “3LS-Authenticate” is a user-friendly mobile phone application, composed of three security layers: activation, cryptography, and authentication. The application focuses on capturing the challenging QR code, in order to generate and send a cryptography-based response to the server for client verification. In case of stolen mobile phones, the “3LS-Authenticate” activation layer only authorizes legitimate owners to use the application.
- **Exclusive QR code decryption.** Although encrypted QR code can be scanned by any QR code-scanning application, “3LS-authenticate” is the only mobile phone application that can decrypt the QR code contents, due to the embedding of the client’s SSL-certificate in the mobile phone.

## III. “3LS-AUTHENTICATE” APPLICATION ARCHITECTURE

Before discussing the “3LS-Authenticate” application architecture, the MUAS challenge-generation phase is presented briefly, in order to show the background on which the application operates.

In the challenge-generation process, whenever a client initiates an SSL/TLS connection with a merchant through their

browser, they jointly establish a shared secret key  $K_{(B,S)}$ . Both client and server exchange freshly generated encrypted-cryptographic numbers (nonce), denoted by  $nb$  and  $ns$ , respectively, to secure exchanged messages. The client is requested to submit the mobile phone number that will be used for OTP authentication. After submitting the received SMS-based OTP, the server composes and appends QR code contents, which contains the server's certificate (CERTs), session server-nonce ( $ns$ ), submitted mobile phone number ( $mn$ ), and response end-point (URL). The contents are then encrypted with the client's public key, as certificates have been exchanged during SSL/TLS protocol establishment. Finally, the QR code image is generated and displayed within the browser, to be captured by the mobile phone application ("3LS-Authenticate").

In the challenge-generation phase, a record of the client is created and added to the server's "Client-Table". This table contains the fresh server-nonce ( $ns$ ) as its primary-key, nonce-hashing result, submitted mobile phone number, QR code contents, and the OTP.

The proposed "3LS-Authenticate" mobile application is responsible for capturing the generated QR code from the web browser and generating a response to be sent to the server for ensuring the client's authenticity.

The "3LS-Authenticate" application is composed of three layers: activation, cryptography, and authentication.

#### 1. Activation Layer

The activation layer is the first layer of the "3LS-Authenticate" application, which resides in the client's mobile phone. The purpose of this layer is to display an alert view that requests the client to submit the Login ID and password, as a requirement for the application to operate.

#### 2. Cryptography Layer

The cryptography layer is the second layer of the "3LS-Authenticate" application. It takes place after capturing the QR code. This layer is based on the client's installed digital certificate in the mobile phone, for cryptography-related functions. It decrypts and splits the QR code contents to recover the server-side nonce. This nonce is hashed and appended with the original value to be encrypted with the server's public key, by employing RSA cryptography.

#### 3. Authentication Layer

The Authentication layer is the third and final layer of the "3LS-Authenticate" application. Its actions take place between the mobile phone application and the server. The authentication layer establishes a secure SSL/TLS communication channel between the mobile phone application and the server. The former sends the cryptography layer's encrypted outcome to the server. The latter decrypts the received message, and compares the recovered server-side nonce against its Client-Table records. If found, the server verifies the recovered hash against its own nonce-hashing result, which exists within

the Client-Table. Otherwise, the authentication process terminates. Finally, the server sends the authentication results to the mobile phone.

The authentication layer also employs a freshly generated nonce for both parties, to be exchanged along with the sent and received messages, in order to prevent replay attacks and avoid impersonators.

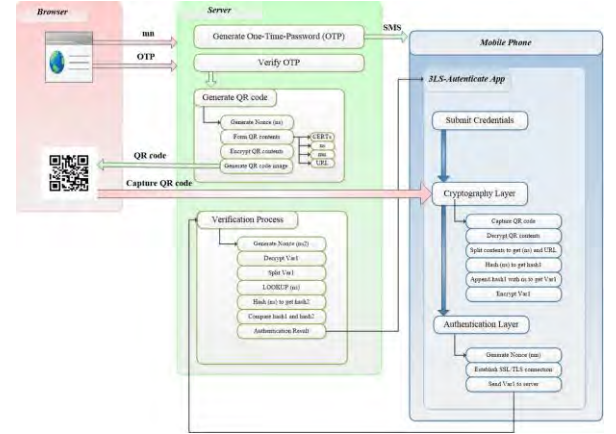


Figure 2. "3LS-Authenticate" Architectural Overview within the MUAS

### IV. "3LS-AUTHENTICATE" ALGORITHM

In this section, we present the core algorithm of the "3LS-Authenticate" application, used when communicating with the server to finalize the client authentication process within the MUAS. The basic workflow is shown in Fig. 2, with a summary of notations used described in Table 1.

Table 1. Summary of notations used in Fig. 2.

Symbol	Description
$nb$	Fresh browser nonce
$ns$	Challenge-Generation Fresh Server nonce
$mn$	Mobile Phone Number
OTP	One-Time Password
CERTs	Server's Certificate
URL	Response End-Point
$B\_id$	Browser ID
$S\_id$	Server ID
$K_{(B,S)}$	Shared Session Key (browser and server)
$K_{(M,S)}$	Shared Session Key (mobile phone and server)
$pk(C)$	Client's Public Key
$sk(C)$	Client's Secret Key
$pk(S)$	Server's Public Key
$sk(S)$	Server's Secret Key
$ns2$	Response-Generation Fresh Server nonce
$nm$	Mobile Phone nonce

The following steps along with Fig. 3 illustrate the “3LS-Authenticate” application sequence of events:

1. The mobile phone user submits their credentials to activate the application.
2. The application captures the QR code with the mobile phone camera and decrypts its contents, using the client’s secret key  $sk(C)$ .
3. QR code contents are split, to recover the server-nonce (ns) and URL.
4. A hashing algorithm is performed on the recovered server-nonce (ns), to obtain hash1.
5. The hash result (hash1) is appended with the original value (ns), and encrypted with the server’s public-key  $pk(S)$ , to obtain Value1.
6. A fresh mobile phone nonce is generated (nm) by the mobile phone application.
7. Using the QR code recovered URL, an SSL/TLS communication channel is established between the mobile phone application and the server.
8. The generated mobile phone nonce (nm), along with Value1, are encrypted with the shared secret-key  $K_{(M,S)}$ , and sent to the server for verification.
9. As the response-generation process represents a new and separate connection from the challenge-generation process, the server generates a new nonce (ns2).
10. The server decrypts the received message from step 8, using the shared secret-key  $K_{(M,S)}$  and by decrypting Value1 (using the server’s secret-key  $sk(S)$ ).
11. The server split the message contents to recover the server-nonce (ns) from the challenge-generation process.
12. The server LOOKUP recovers (ns) against the server’s Client-Table, to match the received response from the mobile phone to the specific client who initiated the process. If the client is found within the Client-Table, then:
13. The server compares the recovered hash1 against hash2. If both match, they represent the server’s own hashing-result (hash2) that took place within the server. Next, the authentication result is generated.
14. The server sends the authentication result to the mobile phone, along with the recovered and received mobile-nonce (nm) and the new server-nonce (ns2).
15. The mobile phone sends its identity (M), along with the received server-nonce (ns2) to the server (S). Otherwise, the server terminates the process.

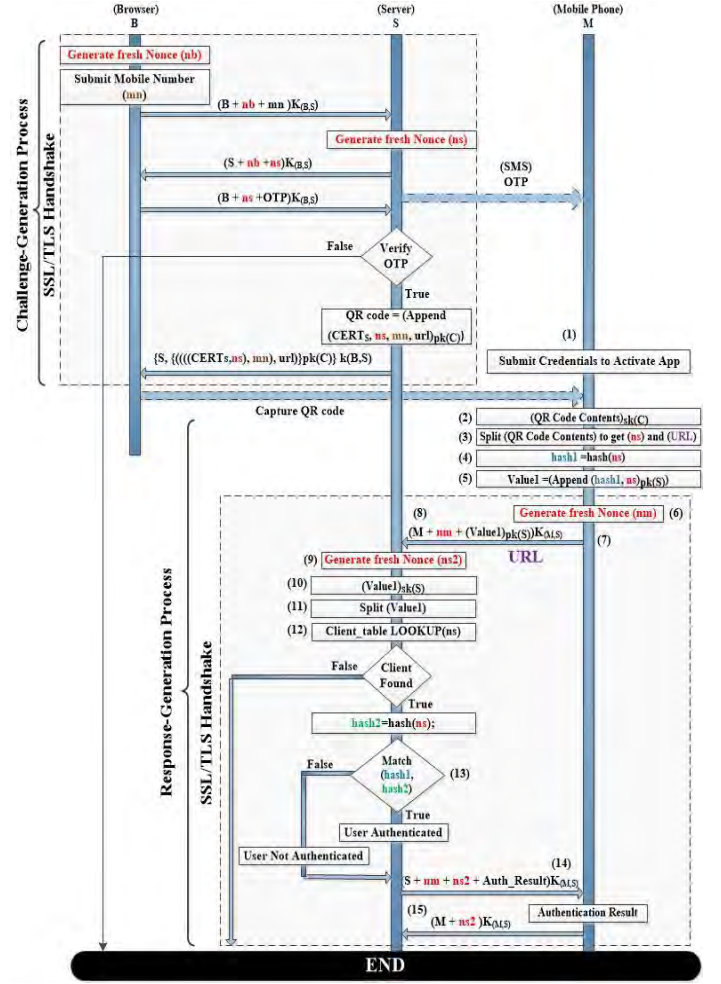


Figure 3. “3LS-Authenticate” workflow within the MUAS

## V. RESPONSE-GENERATION PROTOCOL VERIFICATION

The response-generation protocol finalizes the MUAS authentication process. It takes place between the server (merchant) and the client (mobile phone application), within a secure SSL/TLS communication channel. Hence, a shared secret-key  $K_{(M,S)}$  is established between the two parties. The mobile phone application is responsible for generating a response from the captured-challenge, to be sent to the server for verification. Fig. 4 depicts the MSC of the response-generation protocol, and illustrates the roles of this protocol, which are the server and mobile phone, denoted by  $S$  and  $M$ , respectively.  $S$  knows its own secret-key  $sk(S)$ , and the public-key of the client’s mobile phone  $pk(M)$ . Symmetrically,  $M$  knows its own secret-key  $sk(M)$ , and the public-key of the server  $pk(S)$ .  $M$  is designed to share the same digital certificate with the browser ( $B$ ), where they both represent the same client. Therefore, the client’s public-key  $pk(C)$  is used within the browser role ( $B$ ), and is the same public key used by  $M$ . As a result,  $pk(C)$  represents the same client that uses the browser and the mobile phone. The same process applies for the client’s secret key, which is denoted by  $sk(C)$ .



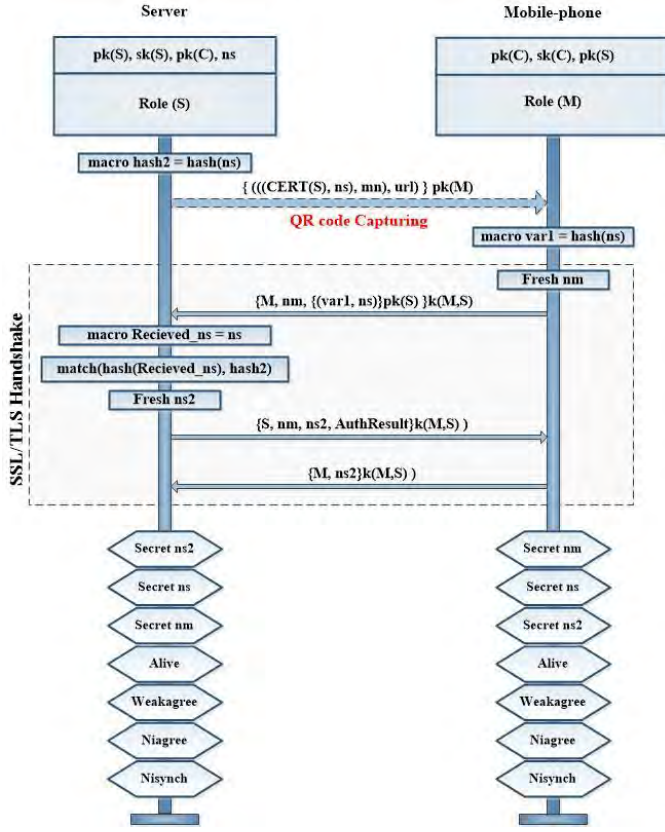


Figure 4. Response-Generation Protocol

The response-generation protocol takes place following capturing and decryption of the QR code. The detailed response-generation (R\_G) protocol functions as follows:

**Step R\_G1:** the mobile phone performs a concatenation of the result, by hashing the recovered server-nonce  $hash(ns)$ , and its original value ( $ns$ ), encrypted with the server public-key.

$$\begin{aligned} macro\ var1 &= hash(ns); \\ \{var1, ns\}pk(S); \end{aligned}$$

The encrypted value is then sent to  $S$ , along with a freshly generated nonce ( $nm$ ), and the mobile phone identity ( $M$ ). All of these attributes are encrypted with the shared secret-key  $K(M,S)$ .

$$send\_1(M, S, \{M, nm, \{var1, ns\}pk(S)\}k(M,S));$$

**Step R\_G2:**  $S$  generates a new nonce ( $ns2$ ), and decrypts the received message from Step R\_G1, and uses LOOKUP to find the recovered ( $ns$ ) against its Client-Table. If found, the process moves to StepR\_G3. Otherwise,  $S$  terminates the process without authenticating the client.

**Step R\_G3:**  $S$  verifies the received ( $ns$ ) from Step R\_G1 through hashing and comparing the result against its own hash result  $hash2$ , in order to match the response generated by the mobile phone with the hashed server-nonce ( $ns$ ), which was generated for the same client who initiated the connection with  $S$  in the challenge-generation phase.

$$macro\ hash2 = hash(ns);$$

$$match(hash(Received\_ns), hash2);$$

**Step R\_G4:**  $S$  sends the recovered mobile-phone nonce ( $nm$ ) back to  $M$ , along with its new nonce ( $ns2$ ), server identity ( $S$ ), and the authentication result, encrypted with the shared secret-key  $K(M,S)$ .

$$send\_2(S, M, \{S, nm, ns2, AuthResult\}k(M,S));$$

**Step R\_G5:**  $M$  decrypts the received message from Step R\_G4, and sends the received server nonce ( $ns2$ ), along with the mobile phone name ( $M$ ) back to  $S$ , encrypted with the shared secret-key  $K(M,S)$ .

$$send\_3(M, S, \{M, ns2\}k(M,S));$$

Prior to the response-generation protocol, a number of internal computations are performed within the mobile phone application ("3LS-Authenticate"), as discussed in section IV. These computations involve decrypting the captured QR code and splitting its contents, in order to recover its information. This represents the essence of the authentication mechanism. Following the last step, control returns to the server, to ensure that the authentication result has been received by the mobile phone.

## VI. SECURITY ANALYSIS

The desire to overcome Man-In-The-Browser (MITB) attacks has caused the MUAS protocol to transition from an insecure network to another communication channel, resulting in the emergence of two separate protocols: challenge-generation and response-generation. Challenge-generation represents the basis on which the protocols are isolated from one another, due to the employment of the QR code as an authentication method. As QR codes within online authentication frameworks can only be processed by mobile phones, moving to another communication channel was a turning point in being able to separate the protocols. Even though both protocols are isolated and verified separately, they are still tightly connected via the QR code contents.

The challenge-response protocol design was based on establishing a secure communication mechanism between participants. Therefore, both protocols were built on top of the SSL/TLS, in order to ensure a secure message exchange. Furthermore, freshly generated nonce were employed between participants in both protocols to satisfy secrecy and authentication security properties, and to help secure exchange messages from impersonators and malicious clients trying to replay the same request. The client nonce gives the client additional protection by preventing malicious users from impersonating the server, and the server nonce helps prevent malicious users from obtaining sent packets, which can be resent later using a separate connection to impersonate the client. Moreover, the nonce prevents adversaries from participating in other sessions by using an intercepted nonce, since it can only be used once.

Another aspect that was considered in the challenge-generation protocol design was to generate a private and secure QR code with unique and useful contents. Uniqueness

was satisfied by including the server's nonce ( $ns$ ) within the QR code contents used for verification within the response-generation protocol. Usefulness is considered in a scalable system, where the server-nonce ( $ns$ ) is used to link the received response from the client's mobile phone to the client initiating the authentication process with the server. A further attribute considered while creating QR code contents was the server's URL. This URL is sufficient to establish a secure connection between the mobile phone application and the server, in order to send the generated response from the mobile phone application to the server for verification.

The proposed "3LS-Authenticate" mobile application was developed with a secure authentication system in mind. Therefore, three levels of security are performed on the mobile phone. The first level requires the client to submit their credentials, which safeguards the mobile phone user's legitimacy, while protecting the server from unnecessary overheads. Cryptographic hash functions are considered an important building block in information security over the internet. Therefore, the second security level employs the (SHA256) hash function, in order to secure messages that will be sent to the server, which in turn satisfy integrity. SHA256 was chosen due to its high speed and reasonable digest size. The third and final level establishes a secure communication channel between the application and the server, in order to send the response to the server, where the authentication result is stored. Furthermore, cryptographic nonces are exchanged between participants to prevent replay attacks and other kinds of malicious acts.

In terms of cryptography, RSA cryptography is considered one of the most secure asymmetric algorithms for this purpose, as it is suitable for real-world use, where secret keys do not have to be shared.

#### a Evaluation Criteria

Employing mobile phones as an authentication device within an authentication system helps satisfy their mobility and usability features, while eliminating the need for special hardware devices. The proposed "3LS-Authenticate" mobile application provides a simple and convenient challenge-response protocol for capturing a QR code in a secure authentication system. The application is user-friendly, due to its minimal requirement for user input.

The proposed system satisfies the following e-commerce security requirements:

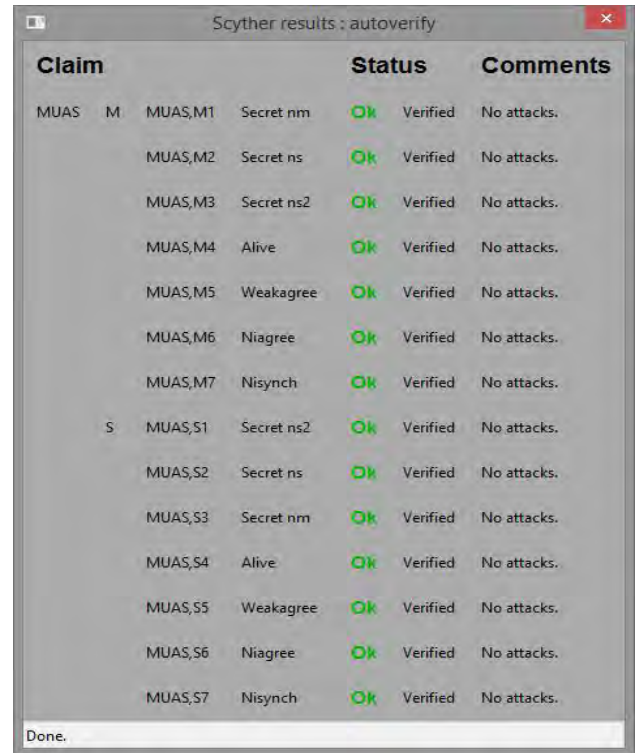
- 1) *Authenticity*. Mutual authentication takes place between the "3LS-Authenticate" mobile application and the server. This is due to the establishment of the SSL/TLS protocol within the response-generation process. In addition, the server identity can be verified through the certificate authority (CA) who issued the digital certificate.
- 2) *Integrity*. The "3LS-Authenticate" application ensures integrity by establishing the SSL/TLS communication channel. In addition, it employs cryptography and QR code contents, and verifies these contents within another communication channel.

- 3) *Confidentiality*. Using cryptographic nonce by both participants within the established SSL/TLS protocol, along with RSA cryptography, helps to secure exchanged messages within the response-generation process from impersonators and other malicious users trying to replay the request. Furthermore, by forming and encrypting substantial information into a QR code, which can only be decrypted via the legitimate phone holder, the application is able to further satisfy confidentiality requirements.

#### b Verification Results

Research in the formal protocol verification area show that formal verification tools have helped in avoiding standardizing protocols with security defects [16]. This section evaluates and verifies the application's response-generation security protocols with respect to possible attacks, using the Scyther security protocol verification tool [17].

During the protocol verification process, Scyther scans a proof tree for all possible protocol behaviours. When verifying the response-generation protocol, where the size of the proof tree is either bounded by the maximum number of runs or unbounded, we determined that no attacks within the state space [18] [19] took place (Fig. 5).



Claim				Status	Comments
MUAS	M	MUAS,M1	Secret nm	OK Verified	No attacks.
		MUAS,M2	Secret ns	OK Verified	No attacks.
		MUAS,M3	Secret ns2	OK Verified	No attacks.
		MUAS,M4	Alive	OK Verified	No attacks.
		MUAS,M5	Weakagree	OK Verified	No attacks.
		MUAS,M6	Niagree	OK Verified	No attacks.
		MUAS,M7	Nisynch	OK Verified	No attacks.
S		MUAS,S1	Secret ns2	OK Verified	No attacks.
		MUAS,S2	Secret ns	OK Verified	No attacks.
		MUAS,S3	Secret nm	OK Verified	No attacks.
		MUAS,S4	Alive	OK Verified	No attacks.
		MUAS,S5	Weakagree	OK Verified	No attacks.
		MUAS,S6	Niagree	OK Verified	No attacks.
		MUAS,S7	Nisynch	OK Verified	No attacks.

Done.

Figure 5. Response-Generation Verification Results

Fig. 5 shows that the secrecy property of  $nm$ ,  $ns$  and  $ns2$  are successfully verified in both roles, and not revealed to an adversary, where  $nm$  and  $ns2$  are nonce-generated by the mobile phone and the server, respectively. The recovered  $ns$  represents the same nonce generated by the server within the challenge-generation protocol. However,  $ns$  is represented as a

constant value within the response-generation protocol. Authentication is also achieved for both roles by satisfying its properties, which are *aliveness*, *weak agreement*, *non-injective agreement*, and *non-injective synchronization*.

Synchronization security is a strong intentional property, requiring all communication messages and their contents to occur exactly as specified by the protocol description, with respect to the order of communicating events. As non-injective synchronization is verified within the protocol, and based on the hierarchy of authentication properties in Fig. 6, any protocol satisfying the synchronization property also satisfies the agreement property [20] [18]. As the non-injective synchronization property also satisfies non-injective agreements, and since every received event in the response-generation protocol is preceded by a send event, the protocol satisfies both properties [18] [21].

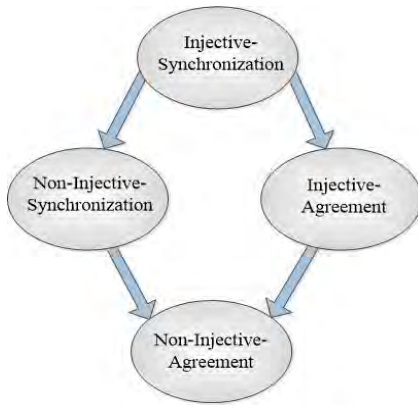


Figure 6. Hierarchy of authentication properties

The agreement property is very similar to the synchronization property, except when placing less restrictions on event ordering. However, most protocols satisfying both properties in practice are considered correct authentication protocols [18].

Moreover, the proposed response-generation protocol satisfies the “alive” property, indicating that the intended communication partner of both participants are alive and have been running the protocol and executing events [18], and that each party has been progressing at least until the last message was sent [22]. Another satisfied property of the proposed protocol is “weak agreement”, which subsumes the “alive” property; each user believes that they are communicating with the intended recipient [22] [23].

## VII. RELATED WORK

Due to recent widespread online attacks and identity theft, a number of studies have analysed new methods for e-commerce authentication. Many of these involve extending the use of mobile phones as SIM-based identity authentication devices.

For example, mobile phones can be used as identity providers (IdP) for Single-Sign-On (SSO) authentication. Some of these require client credentials for logging-in [1] [3]

[4] [24] [25], whilst others necessitate extra hardware devices [2] [26]. Although these approaches satisfy mobility, usability, availability, and security requirements, they also have shortcomings: clients must manage and memorize their credentials; and have the burden of carrying extra devices for authentication, respectively.

Another simple and effective online authentication approach has recently emerged. This approach uses mobile phones to authenticate clients by capturing QR codes, which establishes a secure connection between the desktop, server, and the mobile phone [27]. Some authentication solutions are based on embedding OTPs within QR codes, such as in online banking systems [28] [29] and e-healthcare authentication systems [30]. Another banking system approach prevents attacks from phishing websites, by dividing a bank-generated QR code into two shares. The system sends one share to the user via e-mail, whilst the other is sent through the network to the server, and finally to the user, in order to combine the two shares and obtain the QR code. This is then scanned to recover the embedded OTP [31].

Another approach involves shared secrets between client and server [24] [27] [32]. While Starnberger et al. (2009) use the same QR code methodology, they provide a more secure system, in which the proposed mobile phone application performs cryptography on received data before it is sent back to the server for verification [33]. Moreover, algorithms have increased QR code content security and confidentiality by encrypting QR code contents [34] [35] [36] [37].

Other technologies, such as Bluetooth, are being used with QR code-based authentication schemes to connect a user’s computer with their mobile phone, in order to send a server-generated OTP to the user’s mobile phone, thus increasing security and reducing consumption [37]. QR codes have also been used to develop an Android-based mobile payment system that functions as a wallet [38].

By comparing the previous approaches with the proposed “3LS-Authenticate” application, it was found that our approach can overcome MITB attacks by transitioning from an insecure network (internet) to an OOB communication channel. It is also more secure against malicious attacks and impersonators, due to the employment of secure communication channels and the exchange of freshly-generated nonce between participants.

## VIII. CONCLUSION

In this paper, we have presented the “3LS-Authenticate” mobile application, which plays a major part of a larger system called “Mobile User Authentication System” (MUAS), developed to authenticate online clients through an Out-Of-Band (OOB) communication channel using a mobile phone to capture a QR code as a means of authentication. The system overcomes Man-In-The-Browser (MITB) attacks, by minimizing the use of PC browsers and using mobile phones instead [14].

The proposed “3LS-Authenticate” mobile application is a simple, fast, and secure authentication solution that does not require the input of personal information.

In terms of security, the response-generation protocol design was based on establishing a secure communication



mechanism between participants. Therefore, the protocol was built on top of SSL/TLS, in order to assure secure message exchange. The application's three levels of security ensure an efficient and effective authentication mechanism.

## REFERENCES

- [1] A. Al-Qayedi, W. Adi, A. Zahro, and A. Mabrouk, "Combined Web/mobile authentication for secure Web access control," in *Wireless Communications and Networking Conference, 2004. WCNC. 2004 IEEE*, 2004, pp. 677-681 Vol.2.
- [2] T. Do van, T. Jonvik, T. Do van, and I. Jorstad, "NETp1-09: Enhancing Internet Service Security Using GSM SIM Authentication," in *Global Telecommunications Conference, 2006. GLOBECOM '06. IEEE*, 2006, pp. 1-5.
- [3] G. Me, D. Pirro, and R. Sarrecchia, "A mobile based approach to strong authentication on Web," in *Computing in the Global Information Technology, 2006. ICCGI '06. International Multi-Conference on*, 2006, pp. 67-67.
- [4] A. Tsuyoshi, I. Hiroki, and T. Kenji, "Implementing identity provider on mobile phone," presented at the Proceedings of the 2007 ACM workshop on Digital identity management, Fairfax, Virginia, USA, 2007.
- [5] M. Ashraff, M. L. Kabir, S. M. Aziz, and B. k. Dey, "A Conceptual Framework for a SIM-based Electronic Transaction Authentication System," in *Network and Parallel Computing Workshops, 2007. NPC Workshops. IFIP International Conference on*, 2007, pp. 65-71.
- [6] D. van Thanh, I. Jorstad, T. A. Johansen, E. Bakken, and D. van Thuan, "Pervasive service access with SIM-based VPN," in *Mobile Adhoc and Sensor Systems, 2009. MASS '09. IEEE 6th International Conference on*, 2009, pp. 836-841.
- [7] W. Audun, L. Lars, J. Ivar, and D. Than van, "Secured enterprise access with strong SIM authentication," in *Enterprise Distributed Object Computing Conference, 2006. EDOC '06. 10th IEEE International*, 2006, pp. 463-466.
- [8] T. Do Van, T. Jnvik, I. Jrstad, B. Feng, and T. Do Van, "Strong authentication using dual SIM," in *Intelligence in Next Generation Networks, 2009. ICIN 2009. 13th International Conference on*, 2009, pp. 1-4.
- [9] C. Le-Pong and C. Jyh-Yen, "SIM card based e-cash applications in the mobile communication system using OTA and STK technology," in *Wireless, Mobile and Multimedia Networks, 2006 IET International Conference on*, 2006, pp. 1-3.
- [10] R.-J. Hwang, S.-H. Shiau, and D.-F. Jan, "A new mobile payment scheme for roaming services," *Electronic Commerce Research and Applications*, vol. 6, pp. 184-191, 2007.
- [11] J. T. Issac and J. S. Camara, "An anonymous Account Based Mobile Payment Protocol for a Restricted Connectivity Scenario," presented at the Database and Expert System Applications, 2007. DEXA, 07. 18th international workshop on, 2007.
- [12] A. A. Tabandehjooy and N. Nazhand, "A Lighweight and Secure Protocol for Mobile Payments Via Wireless Internet in M-commerce," in *e-Education, e-Business, e-Management, and e-Learning, 2010. IC4E '10. International Conference on*, 2010, pp. 495-498.
- [13] A. Bottoni and G. Dini, "Improving authentication of remote card transactions with mobile personal trusted devices," *Computer Communications*, vol. 30, pp. 1697-1712, 2007.
- [14] R. Molla, I. Romdhani, W. Buchanan, and E. Fadel, "Mobile User Authentication Systemfor E-commerce Applications," in *International Conference on Advanced Networking, Distributed Systems and Applications 2014, IEEE*, pp. 27-34.
- [15] (1998). *Chilkat Software*. Available: <http://www.chilkatsoft.com/>
- [16] N. Dalal, J. Shah, K. Hisaria, and D. Jinwala, "A comparative analysis of tools for verification of security protocols," *Int'l J. of Communications, Network and System Sciences*, vol. 3, p. 779, 2010.
- [17] C. Cremers, "The Scyther Tool," v1.1.3 ed, April 4, 2014.
- [18] Cas Cremers and S. Mauw, *Operational Semantics and Verification of Security Protocols*: Springer, 2012.
- [19] C. J. F. Cremers, P. Lafourcade, and P. Nadeau, "Comparing State Spaces in Automatic Protocol Analysis," in *Formal to Practical Security*. vol. 5458/2009, ed: Springer Berlin / Heidelberg, 2009, pp. 70-94.
- [20] C. J. Cremers, S. Mauw, and E. P. D. Vink, "Defining authentication in a trace model," in *Fast*, 2003, pp. 131-145.
- [21] C. J. F. Cremers, S. Mauw, and E. P. de Vink, "A Syntactic Criterion for Injectivity of Authentication Protocols," *Electronic Notes in Theoretical Computer Science*, vol. 135, pp. 23-38, 7/5/2005.
- [22] G. Lowe, "A hierarchy of authentication specifications," in *Computer Security Foundations Workshop, 1997. Proceedings., 10th*, 1997, pp. 31-43.
- [23] G. Bella, *Formal correctness of security protocols*: Springer Science & Business Media, 2007.
- [24] K. Choi, C. Lee, W. Jeon, K. Lee, and D. Won, "A mobile based anti-phishing authentication scheme using QR code," in *Mobile IT Convergence (ICMIC), 2011 International Conference on*, 2011, pp. 109-113.
- [25] A. Vapen, D. Byers, and N. Shahmehri, "2-clickauth optical challenge-response authentication," in *Availability, Reliability, and Security, 2010. ARES'10 International Conference on*, 2010, pp. 79-86.
- [26] D. van Thanh, T. Jonvik, F. Boning, D. van Thuan, and I. Jorstad, "Simple Strong Authentication for Internet Applications Using Mobile Phones," in *Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008. IEEE*, 2008, pp. 1-5.
- [27] B. Dodson, D. Sengupta, D. Boneh, and M. S. Lam. (2009). Snap2Pass: Consumer-Friendly Challenge-Response Authentication with a Phone. Available: [senguptas.org/Documents/secure2pass\\_www2009.pdf](http://senguptas.org/Documents/secure2pass_www2009.pdf)
- [28] Y. S. Lee, N. H. Kim, H. Lim, H. Jo, and H. J. Lee, "Online banking authentication system using mobile-OTP with QR-code," in *Computer Sciences and Convergence Information Technology (ICCIT), 2010 5th International Conference on*, 2010, pp. 644-648.
- [29] A. Tandon, R. Sharma, S. Sodhiya, and P. Vincent, "QR Code based secure OTP distribution scheme for Authentication in Net-Banking," *International Journal of Engineering & Technology*, pp. 0975-4024, 2013.
- [30] N. Thiranan and H. Lee, "A Design of e-Healthcare Authentication Framework with QR Code," *International Journal of Security & Its Applications*, vol. 8, pp. 79-86, 2014.
- [31] D. Moholkar, N. Kadam, D. Deokar, A. Kute, and S. Kadam, "An Efficient Approach for Phishing Website Detection using Visual Cryptography (VC) and Quick Response Code (QR code)," *International Journal of Computer Applications*, vol. 115, 2015.
- [32] J. Malik, D. Girdhar, R. Dahiya, and G. Sainarayanan, "Multifactor Authentication Using a QR Code and a One-Time Password," *Journal of Information Processing Systems*, vol. 10, 2014.
- [33] G. Starnberger, L. Frohofer, and K. M. Goeschka, "QR-TAN: Secure Mobile Transaction Authentication," in *Availability, Reliability and Security, 2009. ARES '09. International Conference on*, 2009, pp. 578-583.
- [34] S. Dey, "SD-EQR: A New Technique To use QR Codes in Cryptography," *International Journal of Information Technology & Computer Science (IJITCS)*, vol. 3, pp. 11-21, May/June 2012.
- [35] A. Adsul, A. Shukla, J. Sinojia, R. Sinkar, and S. Jagtap, "Secure Authentication Method using QR Code for Banking," *International Journal*, vol. 3, 2015.
- [36] V. Kale, Y. Nakat, S. Bhosale, A. Bandal, and R. G. Patole, "A Mobile Based Authentication Scheme Using QR Code for Bank Security," *International Journal of Advanced Researchin Computer Science and Management Studies (IJARCSMS)*, vol. 3, pp. 192-196, 2015.
- [37] S. Liu and S. Zhu, "A Novel QR Code and mobile phone based Authentication protocol via Bluetooth," presented at the International Conference on Materials Engineering and Information Technology Applications (MEITA), 2015.
- [38] C. Ugwu and T. Mesigo, "A Novel Mobile Wallet Based on Android OS and Quick Response Code Technology," *International Journal of Advanced Researchin Computer Science and Technology (IJARCT)*, vol. 3, pp. 85-89, 2015.