Received: 25 April 2021 | Revised: 16 May 2021 | Accepted: 17 May 2021
DOI: 10.37917/ijeee.17.1.15 Vol. 17 | Issue 1 | June 2021

Open Access

Iraqi Journal for Electrical and Electronic Engineering
Original Article

An Efficient Mechanism to Prevent the Phishing Attacks

Mustafa H. Alzuwaini*, Ali A. Yassin
Department of Computer science, Education College for Pure Sciences,
University of Basrah, Basrah, 61004, Iraq

Correspondence
*Mustafa H. Alzuwaini
Department of Computer science
Education College for Pure Sciences,
University of Basrah, Basrah, Iraq
Email: pgs2189@uobasrah.edu.iq

Abstract
In the era of modern trends such as cloud computing, social media applications, emails, mobile applications, and URLs that lead to increased risks for defrauding authorized users, and then the attackers try to gain illegal access to accounts of users through a malicious attack. The phishing attack is one of the dangerous attacks caused to access of authorized account illegally way. The finances, business, banking, and other sensitive in states are faces by this type of attacks due to the important information they have. In this paper, we propose a secure verification scheme that can overcome the above-mentioned issues. Additionally, the proposed scheme can resist famous cyberattacks such as impersonate attacks, MITM attacks. Moreover, the proposed scheme has security features like strong verification, forward secrecy, user's identity anomaly. The security analysis and the experimental results proved the strongest of the proposed scheme compared with other related works. Finally, our proposed scheme balanced between the performance and the security merits.

KEYWORDS: Phishing Attack, Scyther, Schnorr Digital Signature, HMAC, Cyber-Attacks, Informal Security Analysis

I. INTRODUCTION

Currently, in the information technology world, several computer systems used the Internet services such as e-payment, e-business, and money exchange[1]. In the Internet era, the applications and systems have developed exponentially and become a colossal point in our lives and activities undeniably unending. Recently, modern information technologies have seen significant hype, but they are suffering from risks like security, malicious attacks, human errors, spam [2]. The security issues consider one of the most important risks faced by information technology and its applications, denote measures put in place to keep information system potentials and services from illegal access. The malicious attack tries to an intentional exploration of computer system and technology-dependent enterprises[3]. The main activities of these attacks steal computer code, change/delete data server, unauthorized access. The most common attacks are cyber-attacks, Man-In-The-Middle (MITM), Social engineering attacks, replay attacks, denial of serve attacks (DoS)[4, 5]. Phishing is a type of cybersecurity attacks employed to steal user's sensitive data like passwords, social security number, credit card numbers, login credentials[6]. The attacker tries to impersonate as a trusted individual by sending a message (ex. text message, email, or instant message) to the victim[7]. This message contains a bogus URL that deceived the victim that looks as if they are coming from a trusted organization, like financial Banks, Universities, e-Bay[8]. The malicious URL can cause the detecting sensitive data of the victim, installation of malware, financial loss for victims, and put the organization's data at risk[9]. Although the security implementation and public awareness are increasing rapidly, the adversaries have abilities to do phishing attacks successfully. Figure 1 explains the unique phishing attacks in the years between 2013–2019 and Fig. 2 shows the phishing attack scenario.

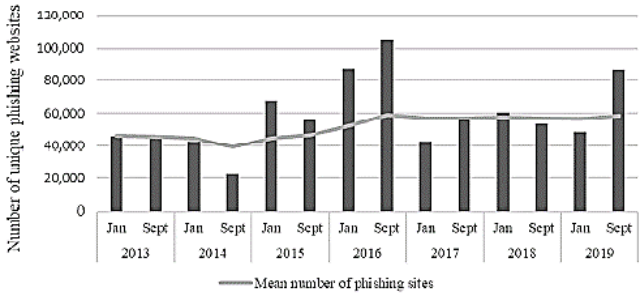


Fig.1: The phishing attacks on the websites between 2013–2019

In this paper, we proposed a verification scheme to prevent well-known cyber-attacks like phishing, MITM, DoS, reply. Additionally, the proposed scheme provides good security merits compared with other related schemes like strong verification, key management, message

unforgeability, an anomaly of user's identity. However, the proposed scheme consists of three main phases: setup phase, registration phase, verification phase. These phases are responsible to manage the exchange of information between main components (User, Authentication server, Community server) in a secure way relied on Schnorr digital signature, HMAC, Levenshtein distance. The security analysis and scyther tool prove that the proposed scheme safe against common attacks [10, 11]. Furthermore, we gain good results in the computational and communicational cost compared with related works.

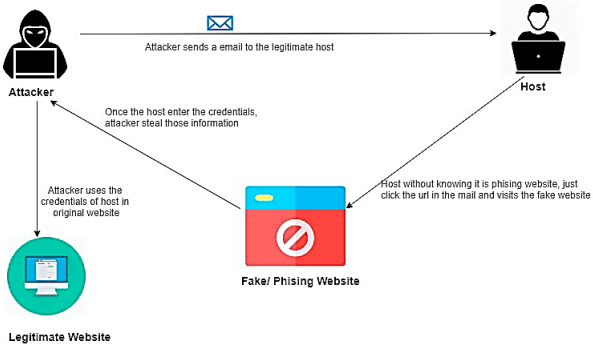


Fig.2: The phishing attacks scenario

The remaining sections of this paper are: *Section 2* describes the main security definitions. *Section 3* describes the primitive tools. *Section 4* shows the literature review. *Section 5* focused on the proposed scheme. *Section 6* views the security analysis and experimental results. *Section 7* denotes the conclusion.

II. PRIMITIVE TOOLS

1. Schnorr digital signature: Schnorr digital signature has been presented to improve ElGamal digital signature by minimizing signature size. It is very useful, qualified, and generates a short signature size[12-14].
This scheme consists of three parts:
KeyGen : The prover chooses two large prims p and q where q is a factor of $p-1; q \geq 2^{140}, p \geq 2^{512}$ and do the following steps:

- Choose an element $g \in \mathbb{Z}_q{}^*$ such that $g^q = 1 \, mod \, p$.
- Select $x \in \mathbb{Z}_q{}^*$, the private key is x and the public key is $y = g^x mod \, p$.
- The public elements are (g, p, q, y).

Sign (g, x, M)**:** According to the input message (M), the private key (x), the concatenation function (||), and the one-way hash function H:$(0,1)^* \in \mathbb{Z}_q{}^*$H: Error! Bookmark not defined.$^* \in Z_q$, the prover performs the following points:

- Choose a random number $k \in Z_q$, and set $r = g^k mod \, p$.
- Calculate the first signature $E = H(M||r)$.
- Calculate the second signature $S = k + xE \, mod \, q$
- Send (M, E, S) to the verifier.

Verify (M, E, S): upon receiving M, S, E and the public elements (g, p, q, y), the verifier performs the following steps

- Set $V = g^S y^{-E}$

$$= g^{k+xE} g^{-xE}$$
$$= g^k$$

- Compute $EV = H(M||V \, mod \, p)$.
- Compare $EV \stackrel{?}{=} E$, if they are matches, the massage accepted; otherwise, it is rejected. Figure 3 explains the mechanism work of Schnnor digital signature

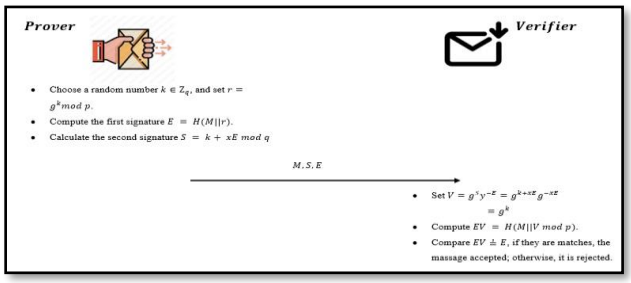


Fig.3: Schnorr signing/verifying

2. *HMAC*: HMAC is a keyed hash function for authenticating the transmitted messages between legitimate parties. It was obtained by running the cryptography hash functions such as (MD5, SHA-1, SHA-256). It is a secure function, attractive choice, efficient, easy to implement[14]. $HMAC$ considers secure against cryptographic attacks and uses multiple parameters such as $ipad, opad$. The value of $opad$ is the block-sized $outer \, padding$, consists of repeated bytes $0x5c$ and $ipad$ is $inner \, padding$, consists of repeated bytes $0x36$.

- ***keyGen***: This function uses the key length (n) as input and runs the $key - generation$ function to obtain s and chooses $S_k \leftarrow (0,1)^n$, where s is the key described the hash function's family.
- $\boldsymbol{Mac_{S_k}(M)}$: Depended on inputs (s, S_k) and message, the HMAC can compute using the following equation $(Eq1)$:

$$t = HMAC^s_{S_k}(M)$$
$$t = H^s_{IV}(S_k \oplus opad || H_{IV}(S_k \oplus ipad || M)) \quad (1)$$

where t is the message tag, then the output of this function is (M, t).

- $\boldsymbol{verify_{S_k}(M, t)}$: if $Mac_{S_k}(M)$ is equal to t then accepts M else rejects it. Figure 4 explains the steps of HMAC.

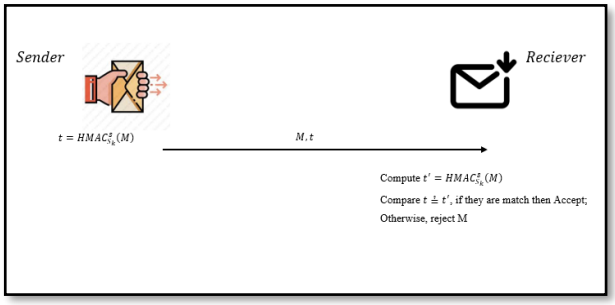


Fig.4: HMAC function

3. *Levenshtein Distance (LD):* The Levenshtein distance is a measure to determine the number of differences between two messages[15]. Assume, we have m1=" network" and m2=" networks", the Levenshtein distance between m1 and

m2 is 1. The mechanism of computing *LD* between two strings *a* and *b* using the following equation (*2*):

$$\mathrm{lev}(a,b) = \begin{cases} |a| & \text{if } |b| = 0, \\ |b| & \text{if } |a| = 0, \\ \mathrm{lev}(\mathrm{tail}(a), \mathrm{tail}(b)) & \text{if } a[0] = b[0] \\ 1 + \min \begin{cases} \mathrm{lev}(\mathrm{tail}(a), b) \\ \mathrm{lev}(a, \mathrm{tail}(b)) \\ \mathrm{lev}(\mathrm{tail}(a), \mathrm{tail}(b)) \end{cases} & \text{otherwise.} \end{cases} \quad (2)$$

III. Related Works

Huang et al.[16] developed an approach to avoid online phishing attacks by routing visitors to legitimate web pages based on site signature merged between image and text-based attributes and used it to extract the real URL. Their approach has suffered from the encumbrance of signature construction allowed to phisher by applying his attacks in the case of the web page getting missed.

Bojjagani et al. [17] proposed an authentication protocol to prevent phishing attacks applied in the mobile payment system using *Elliptic Curve Digital Signature Algorithm* (ECDSA). Their protocol depends on an authentication server to send a nonce message to the user and check the authority of the user's signed data. After that, an authentication server sends signed context information to the genuine bank for avoiding phishing attacks[18]. The limitations of this protocol are that the attacker mimics the behavior of the protocol by sending a fake link to the user and then sends a nonce to delude the user to be coming from the authentication server. Additionally, this work suffers from an anomaly of user's identity, resisting of impersonate attack and needs strong verification.

Roy et al. [19] proposed a scheme to use mobile-based authentication in cloud computing. In this scheme, they presented a universal subscriber identity module (USIM) depends on the identity verification method. The USIM is used as a primary identity to begin the authentication progression. But in the case of the device get stolen, authentication will get uselessly, and the entire process will get canceled [20].

Lin et al [21] introduced a secure scheme in the smart learning application in the cloud environment. Their scheme registers the user based on his ID to the authentication server. In their scheme, the user sends the hash of the password to the authentication server in the symmetrically encrypted form. The authentication server decrypts and can get the hash value of the password. Their scheme was secure against the Man-In-The-Middle (MITM) attack, but their scheme vulnerable to the phishing attack because of the password sharing between the communication entities.

Rose et al [22] proposed a scheme called password hash ($PwdHash$) that can generate different passwords for each website when the user uses the same password for all these websites. If the user's password was phished from an attacker, he can simply use it for all websites related to the user. Their scheme aims to use $PwdHash()$ for making the plain password is probabilistic. The mechanism of works achieves by applying $pwd' = pwdHash(pwd, seed)$ and sends the pwd' instead of pwd to the selected website. The value of $seed$ is a unique parameter related to the website's domain name. Therefore, the attacker cannot phish the user's password to make illegal access to the user's websites.
Although the proposed scheme works perfectly, the attacker's goal is not to steal the user's password, he aims to steal the user's confidential data when he visits the attacker's website instead of a real website.

Munivel et al. [23] proposed an authentication scheme to provide security in the mobile cloud environment, this scheme contains three parties (cloud user (*U*), cloud service provider (*CSP*), and trusted third party (*TTP*)). Their scheme has three phases. The first one creates a group called *G* and its members. The TTP shares the elements of the group with the communication entities. The second one handles the registration of *U* and *CSP* with the *TTP*. The last one checks the authority of *U* and *CSP* to achieve mutual authentication. Their scheme needs a strong verification because the *CSP* sends a nonce in a plain form that leads to a phishing attack by simulating the *CSP* behaviors. The attacker can impersonate the *CSP* and sends a nonce to *U* to redirect him to his malicious server.

Lee et al.[24] proposed an authentication scheme for the smart-learning system that is securely operated in the cloud computing environment. They depended on a two-factor authentication scheme to achieve privacy and safety for the learners.
This scheme fails against a phishing attack, impersonate attack, device stolen attack.

Okunoye et al.[25] developed an anti-phishing approach using an advanced heuristic technique. In this approach, when a fishy website was discovered, the blacklist was immediately updated by inserting the website's domain name into this list. If a valid website is identified, the white list will update in the same manner. Additionally, when the user visits a website, this approach was first checked if the website was a phishing website or not and given permission to access the same website accordingly. This approach suffers from user privacy and needs security analysis against well-known cyberattacks such as MITM, reply, impersonate, insider. By the way, this approach requires more time in the computation/communication cost because it is needing access to the database for each login phase.

In this paper, we propose a strong verification scheme based on digital signature and HMAC function. The proposed scheme has many good metrics such as anomaly of user's identity, key management and resists the famous attacks like impersonate attacks, phishing attacks, replay attacks, DoS attacks, and device stolen attacks. Our proposed scheme was verified formally with scyther and informally by using cryptography proofs. Additionally, we obtain a good result in the security analysis and computation/communication cost compared with the related works. The main comparisons are shown in table 1 below.

TABLE 1
COMPARISON WITH OTHER RELATED WORKS

Property	[16]	[17]	[19]	[21]	[22]	[23]	[24]	[25]	Our
User's identity anomaly	NO	NO	YES	NO	NO	YES	YES	NO	YES
Forward secrecy	NO	YES	YES	YES	NO	YES	YES	NO	YES
Login phase efficiency	NO	YES	YES	YES	YES	YES	YES	NO	YES
Strong verification	NO	NO	YES	NO	YES	NO	NO	NO	YES
Public IP verification	NO	NO	NO	NO	NO	NO	NO	NO	YES
Resist against phishing attack	NO	NO	NO	NO	YES	NO	NO	YES	YES
efficient phishing URL detection	NO	NO	NO	NO	NO	NO	NO	YES	YES
Resist against message unforgeability	YES	YES	NO	NO	NO	YES	NO	NO	YES
Resist against MITM attack	NO	YES	NO	NO	YES	NO	YES	NO	YES
Resist against Replay attack	YES	YES	YES	YES	YES	YES	YES	NO	YES
Resist against DoS attack	YES	YES	NO	YES	YES	YES	YES	NO	YES
Resist against insider threat	YES	YES	YES	YES	YES	YES	NO	NO	YES
Using secure index file	NO	NO	NO	NO	NO	NO	NO	YES	YES
Resist against traffic analysis attack	NO	YES	YES	YES	YES	YES	NO	NO	YES
Resistant to Stolen user's device Attack	NO	NO	NO	YES	NO	YES	NO	NO	YES
Formal verification with scyther	NO	YES	NO	NO	NO	YES	NO	NO	YES

IV. THE PROPOSED SCHEME

Our work presents a good authentication protocol to avoid Man-In-The-Middle (MITM), Social engineering, and Phishing attacks. The proposed scheme consists of three major components: User (U_i), Community Server (C_i), Authentication Server (AS). Additionally, there are three main phases: The setup Phase, Registration Phase, and Verification Phase. The proposed scheme is shown in Fig. 5 bellows. The user wishes to interact with the authorized community server such as University, Bank, Hospital based on his/her device, included sensitive data like information about the user's credential card. The community's server deals with real user (U_i) via official web pages. The community's server redirects the user's request to an authentication server for verifying his request and the community's Domain Name System (DNS). The authentication server generates and sends the encrypted Verification Code (VC) to the user (U_i). Finally, the user checks the validity of an authentication server using VC for completing the user's request. Additionally, our proposed scheme creates $secure\ index\ file(SIF)$ contained valid URLs of authorized communities to prevent phishing and social engineering attacks.
The main phases are:

1. *Setup Phase:*

1. In the setup phase, AS sets $Secure\ index\ file\ (SIF)$ contained all the valid $URLs$ for the authorized communities.
2. AS chooses two large prims p and q.
3. AS chooses a generator g of the q order.
4. AS generates public and private keys of $Schnorr\ Digital\ Signature$ as follows:

- AS selects $x_{AS} \in \mathbb{Z}_q{}^*$, the private key is $SK_{AS} = x_{AS}$ and the public key is $PK_{AS} = g^{SK_{AS}}\ mod\ p$.

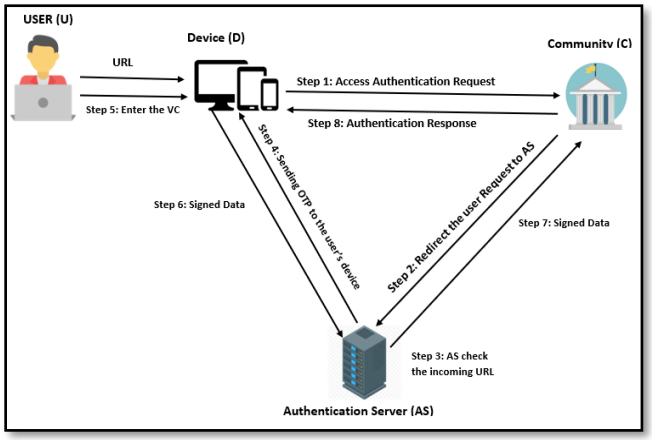


Fig.5: The proposed scheme

Table 2 shows the main abbreviations used in the proposed scheme.

TABLE 2
NOTATION USED IN THE PROPOSED SCHEME

Character	Description
UI	User information that's needs to access the community site
U_i	The user
ID_i	User's unique Identity
C_i	Community Server
$C_{i_{id}}$	Community identity
$C_{i_{URL}}$	Community's domain name
AS	Authentication Server
$S_{k_{U_i}}$	User's shared key
PK_{U_i}	The public key of the user
SK_{U_i}	The private key of the user
PK_{AS}	The public key of the Authentication server
SK_{AS}	The private key of the Authentication server
PK_{C_i}	The private key of the community server
SK_{C_i}	The private key of the community server
LD	*Levenshtien Distance*
SIF	Secure index file
T_i	Time Stamp
DNS	Domain name system
M_i	Communication Message
f_i	Authentication flag
$\mathbb{Z}_q^*$	Group of q order and the elements are relatively prime with the order of group
$\|\|$	Concatenation function
VC	Verification code

2. *Registration Phase*

2.1. *Community Registration Phase*

1. The C_i should register its information to AS such as community identity ($C_{i_{id}}$), Domain name $(C_{i_{URL}})$.
2. Upon receiving the information of C_i, AS compares $C_{i_{URL}}$ with the SIF depended on $Levenshtein\ distance(LD)$ if $LD \notin [1..3]$, AS rejects this community; Otherwise, he inserts the $C_{i_{URL}}$ to the SIF.
3. AS generates public and private keys of $Schnorr\ Digital\ Signature$ to the C_i as follows:

- AS chooses $x_{C_i} \in \mathbb{Z}_q^*$, the private key $SK_{C_i} = x_{C_i}$ and the public key is $PK_{C_i} = g^{SK_{C_i}} \bmod p$ and sends these keys to the C_i.

Figure 6 explains the steps of C_i registration phase.

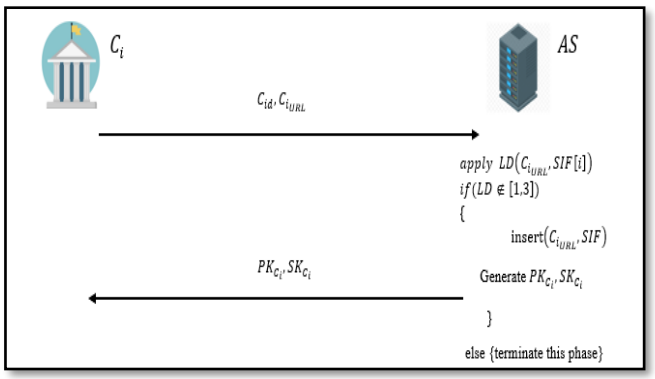


Fig.6: Community Registration phase

2.2. *User Registration Phase*

1. The user U_i registers his information (Identity (ID_i), Personal Identity Number (PIN_i) into C_i. After that, C_i checks the ID_i in his database, if not exist go to step 2; Otherwise, terminate the current phase.
2. The C_i forwards the ID_i to AS for obtaining the private and public keys to the user. After that, AS generates $x_{U_i} \in \mathbb{Z}_q^*$, the private key is $SK_{U_i} = x_{U_i}$ and the public key is $PK_{U_i} = g^{SK_{U_i}} \bmod p$. Additionally, AS computes shared key $(S_{k_i} \in \mathbb{Z})$ and then computes anomaly of user's identity $ID'_i = HMAC_{S_{k_i}}(ID_i || C_{i_{id}})$.
3. AS sends (S_{k_i}, PK_{U_i}, SK_{U_i}) to the U_i and sends ID'_i to C_i. Figure 7 shows the steps of the registration phase.

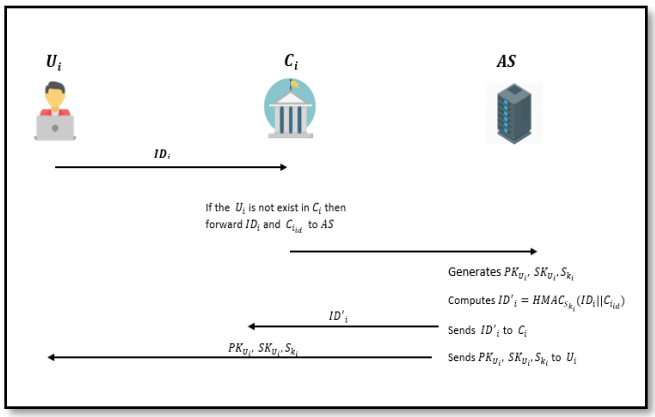


Fig.7: User Registration phase

3. Verification Phase

In this phase, U_i applies the following steps when he receives a certain URL via his/her email or any social media applications and then clicks on the URL of his community for using the services and facilities of the URL. However, U_i sends $M_1 =< ID_i, URL >$ to C_i.

$$C_i \leftarrow U_i : M_1$$

1- Upon receiving the user's information M_1, C_i checks the identity of U_i, if holds; C_i sends $M_2 =< M_1, C_{i_{id}}, ID'_i >$ to AS over a secure connection channel. $AS \leftarrow C_i : M_2$.

2- Upon receiving M_2, AS sets $f_i = 0$ and then restore the URL from M_2 to compare with the SIF depending on $Levenshtein\ distance(LD) = \forall j\ LD(URL, SIF[j])$, If $LD \notin [1..3]$, AS sets $f_i = 1$. After that, he checks the authority of U_i by comparing ID'_i, if the result doesn't match then AS terminates the current phase; Otherwise, AS generates v$erification\ code(VC)$ that consist of four digits. Additionally, AS encrypts VC using $E = Enc_{S_{k_i}}(VC)$ then sends $M_3 =< E, AS_{id}, C_{id} >$ to the U_i.

$$U_i \leftarrow AS : M_3.$$

3- Upon receiving M_3, the U_i perform the following steps:

- Set $UI =< URL,\ IP,\ C_{id}, \ldots etc >$.
- Compute $VC' = Dec_{S_{k_i}}(E)$ and $M_4 = (UI\ ||VC')$.
- Choose $k_1 \in \mathbb{Z}_q^*$ and set $r_1 = g^{k_1}$.
- Compute $e = H(r_1 || M4)$ and $s = k_1 + SK_{U_i} * e$

➢ Send $Sign_{SK_{U_i}}(M_4) =< M_4, s, e >$ to the AS.

4- AS verifies whether the UI is correct or not based on the received s, e by running $Verify_{PK_{U_i}}(M_4, s, e)$.

- Set $r_1' = g^s y^{-e}$.

$$r_1' = g^{k_1 + SK_{U_i} * e} * g^{-SK_{U_i} * e}.$$
$$r_1' = g^{k_1}.$$

- compare $e \stackrel{?}{=} H(r_1' || M_4)$ if they are equal then go to the next step; Otherwise terminate this phase.
- Compare $VC \stackrel{?}{=} VC'$, if they are a match, AS informs C_i via a signed message contained $M_5 = (ID_i || C_{i_{id}} || T_1 || f_i)$ where T_1 is a duration time of the verification process, AS executes the following steps:

✓ Choose $k_2 \in \mathbb{Z}_q^*$ and set $r_2 = g^{k_2}$

✓ Compute $e' = H(r_2 || M_5)$ and $s' = k_2 + SK_{AS} * e'$.

$$C_i \leftarrow AS: Sign_{SK_{AS}}(M_5) =< M_5, s', e' >.$$

5- In the time T_2, C_i receives M_5 and the signature (s', e') then C_i perform $Verify_{PK_{AS}}(M_5, s', e')$ as follows:

- Set $r_2' = g^{s'} y^{-e'}$.

$$r_2' = g^{k_2 + SK_{AS} * e'} * g^{-SK_{AS} * e'}.$$
$$r_2' = g^{k_2}.$$

- compare $e \stackrel{?}{=} H(r_2' || M_5)$, if they are equal C_i perform the next step; Otherwise terminate this session.

Restore f_i from M_5 and compare $f_i \stackrel{?}{=} 1$, if they are equal then C_i extract T_1 to compare with T_2 to determine the validity of login duration time, if $(T_1 - T_2 \leq \Delta T)$, C_i response to the U_i. Otherwise, C_i terminates the current session. The verification steps are states in Fig. 8 below:

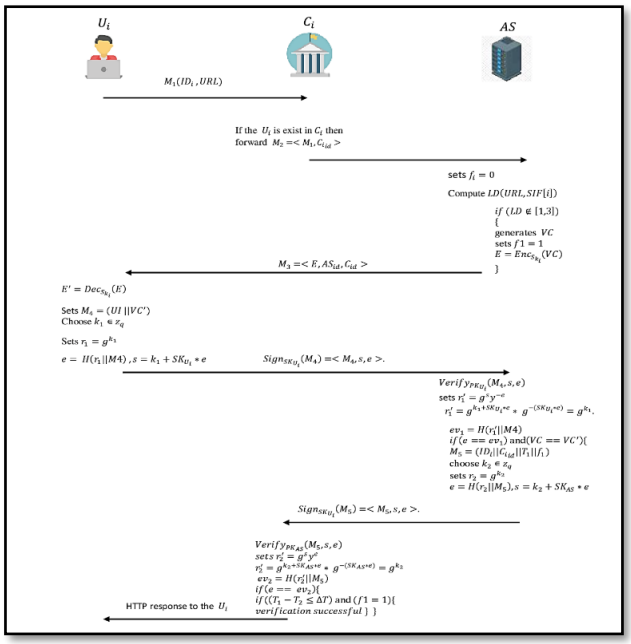


Fig.8: Verification Phase

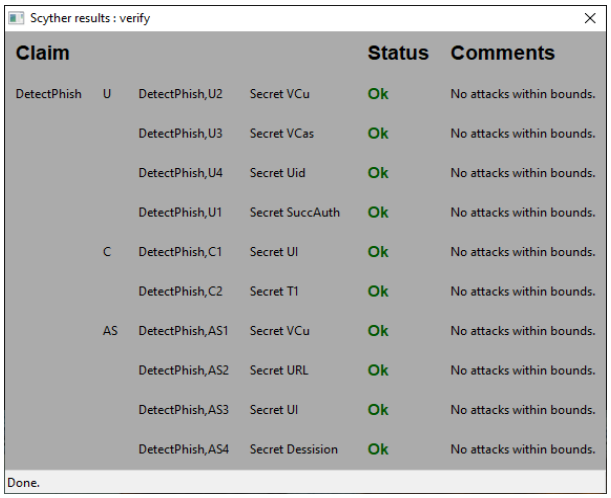
V. SECURITY ANALYSIS

Currently, we pay more attention to proving the proposed scheme that can resist significant attacks such as Phishing attacks, Man-In-The-Middle (MITM), Replay attacks, Insider attacks. Additionally, our work has several security merits like strong verification, session key agreement, forward secrecy, and data integrity. We analysis the proposed scheme the formal and informal as follows:

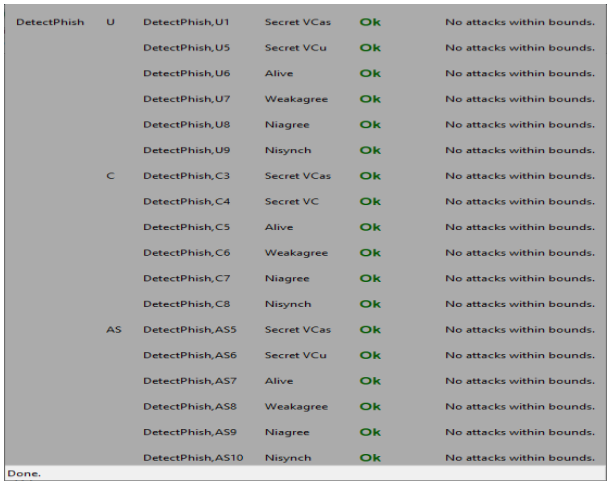
1. *Formal Security Analysis with scyther tool*

Scyther is a vital tool for formal security analysis that works under certain conditions: an attacker should be known the decryption key to achieving the plaintext of the ciphertext. Scyther tool has many advantages: 1) It considers an unbounded model for checking many security schemes (like authentication, verification, access control), 2) It permits the soundness of a proposed scheme for all possible behaviors such as malicious attacks. To implement any suggested scheme, we should be written in the *security protocol description language* (SPDL) which defines protocols/schemes, support expressions for encryption/decryption and signing, and sending/receiving events[10].

We write our proposed scheme using SPDL language and display the results in the case of ($Automatic\ Claim$) and ($Verification\ Claim$). Below the result of our proposed scheme in scyther shown in Fig. 9 (a, b), respectively.



a. Verification Claim



b. Verification auto Claim

Fig.9: Verification results in scyther tool

Table 3 shows the security goals of our proposed scheme in scyther tool.

TABLE 3

THE SECURITY GOALS OF THE PROPOSED SCHEME

Goal 1: *claim U1(U, Secret, VCas)*	**Goal 5:** *claim AS3(U, Secret, Decision)*
Goal 2: *claim U2 (U, Secret, VCu)*	**Goal 6:** *claim C1(U, Secret, UI)*
Goal 3: *claim U4(U, Secret, SuccAuth)*	**Goal 7:** *claim C2(U, Secret, T1)*
Goal 4: *claim AS1(U, Secret, VCas)*	**Goal 8:** *claim AS2(U, Secret, URL)*

2. *Informal Security Analysis*

In this section, we present the ability of the proposed scheme to resist famous attacks such as Phishing, MITM, Insider attacks. Moreover, our work possesses several security features like strong verification, forward secrecy, user identity anomaly in the authentication server, and message Unforgeability[5].

Proposition 1. ***Our proposed scheme supports strong verification property.***

Proof. The secure verification means each component can verify the other based on the secure way. In the proposed scheme, we can notice this feature as follows:

1- U_i sends M_1 to C_i.
2- C_i generates and sends M_2 to AS.
3- AS checks the validity of URL and do the following steps:

Generate VC and encrypts $E = Enc_{S_{k_i}}(VC)$ then Send M_3 to U_i.

4- U_i decrypts E as $VC' = Dec_{S_{k_i}}(E)$ then he constructs a signature message (M_4) and sends to the AS.
5- AS verifies the signed message M_4.

- $Verify_{PK_{U_i}}(M_4, s, e)$.
- If the signature is verified, AS compares $VC \stackrel{?}{=} VC'$; if the results match, he informs C_i via signed message consist of M_5 as follows:

6- C_i performs signature verification on M_5 . $Verify_{PK_{AS}}(M_5, s, e)$.
compare $(T_1 - T_2 \leq \Delta T)$; if the condition holds; C_i response to the U_i; Otherwise C_i terminates the current session.

From the above scenario, we conclude our proposed scheme supports the strong verification property.

Proposition 2. ***The proposed scheme supports forward secrecy***.

Proof. In the proposed scheme, the popular session key relies on ($S_{k_i}, SK_{U_i}, PK_{U_i}, SK_{AS}, PK_{AS}$) used in the verification phase. We notice that an adversary (Ā) fails to restore main keys such as $SK_{U_i}, PK_{U_i}, SK_{AS}, PK_{AS}$ because of the exchanged parameters (s, e, s', e') between components generated once for each verification request. At the same time, Ā cannot compute (r_1', r_2') because Ā fails to obtain the main parameters (SK_{U_i}, SK_{AS}) if we assumed; Ā has the public keys ($g^{SK_{U_i}}, g^{SK_{AS}}$) because it's too hard to obtain (SK_{AS}, SK_{U_i}) from ($g^{SK_{U_i}}, g^{SK_{AS}}$) . This is $desecrate\ logarithmic$ assumption it was proved too hard. Finally, Ā has not the ability to $Sign/Verify$ any message via a communication channel between components. Therefore, Ā cannot impersonate legal user (U_i), server (AS), community (C_i). Therefore, our work has forward secrecy and an adversary cannot apply impersonate attacks.

Proposition 3. ***The proposed scheme supports signature unforgeability***.

In our scheme, the signed messages exchanged between ($U_i \rightarrow AS$) and ($AS \rightarrow C_i$).

We take $U_i \rightarrow AS$ for example to prove that.

Proof. Signature forgery means that the adversary Ā can sign a chosen message to AS for impersonating the legitimate user U_i. The signed message M_4 needs the private key $SK_{U_i}$of the U_i. The Ā knows public parameters (g, q, p) and outputs a message m', as well as, Ā performs the following points:

- Choose a message $m' \in \mathbb{Z}_q^*$.
- Select $k' \in \mathbb{Z}_q^*$.
- Pick $SK'_{U_i} \in \mathbb{Z}_q^*$
- Set $r' = g^{k'} mod\ p$
- Find $e = H(r'||m')$ and $s = k' + SK'_{U_i} * e$
- Finally, send (m', s, e) to AS
  AS performs $Verify_{PK_{U_i}}(m', s, e)$.
  $r_1' = g^s y^{-e}$.
  $r_1' = g^{k'+SK'_{U_i} * e} * g^{-SK_{U_i} * e}$.
  $r_1' = g^{k_1 - e(SK_{U_i}' + SK_{U_i})}$
  Compare $e \stackrel{?}{=} H(r_1'||m')$

The result of the above comparison is a mismatch because of AS fails to compute the valid value of r. As result, AS rejects the message m' and our scheme can support the message unforgeability property.

Proposition 4. ***The proposed scheme supports the authentic message and message integrity***.

Assume that we have an adversary Ā that $eavesdrops$ the messages between the communication entities. The sensitive signed messages exchanged between ($U_i \rightarrow AS$) and ($AS \rightarrow C_i$). In this proof, we show that the adversary hasn't the ability to modify/corrupt any message transmitted between the communication entities.

Proof. The adversary Ā intercepts all the transmitted messages by the communication entities such as the messages between ($U_i \rightarrow AS$) and tries to modify the legal messages by performing the following steps:

- He catches the message M_4 and changes it to M_4' then forwards to AS.
- AS performs $Verify_{PK_{U_i}}(M_4', s, e)$ as follows:
  Set $r_1' = g^s y^{-e}$.
  $r_1' = g^{k_1 + SK_{U_i} * e} * g^{-SK_{U_i} * e}$.
  $r_1' = g^{k_1}$
  Compare $e \stackrel{?}{=} H(r_1'||M_4')$

The comparison of the above condition was not holding because e computes with original message $H(r_1'||M_4)$ and AS computes $H(r_1'||M_4')$. As result, the message was

rejected from AS and our proposed scheme archives the authentic message and message integrity properties.

Proposition 5. ***The proposed scheme supports user's identity anonymity***.

Proof. In the registration phase, the U_i gives his important information such as $(ID_i, name, gender, \ldots. etc)$ to the C_i, after that, C_i forwards $(ID_i, C_{i_{id}})$ to the AS to obtain the main parameters such as $(S_{k_i}, SK_{U_i}, PK_{U_i})$. The AS sends these parameters to the U_i based on his ID_i, then AS saves the ID_i as anomaly form by applying $ID'_i = HMAC_{S_{k_i}}(ID_i || C_{id})$. Finally, the user's record is shown in table 4 below:

TABLE 4.
THE USER'S INFORMATION IN AS

ID'_i	PK_{Ui}	SK_{Ui}	S_{ki}
123wre23232f	25	55	67
5edf533ffdfbf	87	43	44
232dfgb788r4	87	66	99
8yuyj654kklrf4	43	45	22

In the verification phase, U_i sends ID_i, URL to the C_i, then C_i retrieve his anomaly identity (ID'_i) for sends to the AS. The AS known nothing about the user's identity and he retrieves the user's main keys based on ID'_i instead of ID_i. As result, if AS gets hacked, the attacker was unable to determine the user's identity associated with the main keys because the HMAC is a one-way function.

Proposition 6. ***The proposed scheme can resist phishing attacks.***

Proof. Assume the U_i receives an email from the Ā contained a phish URL (www.xcommunity.com) for obtaining the user's sensitive information, the U_i clicks on the appropriate link and visits the Ā instead of C_i server (www.community.com).

Since the U_i needs encrypted credentials from AS, Ā has no choice, he should open a session with the real community C_i.

$$C_i \leftarrow \bar{A}: URL_{\bar{A}}, ID_i.$$

C_i forwards $< URL_{\bar{A}}, ID_i, C_{id} >$ to AS then AS extracts $URL_{\bar{A}}$ to compare with SIF using $levenshtien$ distance, since the distance is located between 1 and 3; AS takes a decision it's an unauthorized server and sets $f_i = 0$. Figure 10 explains the phishing attack prevention.

On the other hand, Ā can easily to mimics the AS and C_i behaviors by sending malicious URL ($URL_{\bar{A}}$) to the U_i and then sends a fake verification code, since the U_i needs to decrypts the VC, Ā cannot mimic the AS behavior because he didn't know the shared key (S_{k_i}). As result, our proposed scheme prevents phishing attacks.

Proposition 7. ***The proposed scheme can resist Replay attacks.***

Proof. Assume the attacker (Ā) interrupts the U_i critical messages that have transmuted between the user and legitimate servers (AS, C_i). The Ā attempts to resend the user's message to the valid destination in the next time. Since the verification phase was determined with $time\ stamp$ (T_1) and random value $r = g^k$, the service provider rejects any requests from this ID_i because the T_1 was exceeded because of the condition $(T_1 - T_2 \leq \Delta T)$ was not hold and the value of r was repeated because the probability of chooses the same random values (k) is *negligible*. Therefore, our proposed scheme resists the replay attack.

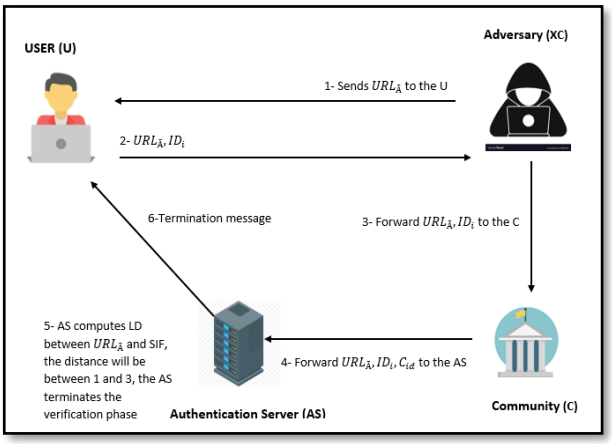


Fig.10: The phishing attack prevention

Proposition 8. ***The proposed scheme can resist DoS and MITM attacks.***

Proof. Assume the attacker (Ā) is presented in the communication channel and tries to steal the sensitive data of U_i that transmuted between legitimate parties (AS, C_i, U_i). The Ā tries to determine the user's main parameters (S_{k_i}, SK_{U_i}) for impersonating the user's ID_i then makes illegal access to the real webserver. since he can't determine the user's main parameters because SK_{U_i} is protected by discreet logarithmic assumption and S_{k_i} is very hard to guess[14]. Additionally, Ā makes a huge of traffic to the C_i for shut down the C_i server to becomes inaccessible, since C_i received the requests multiple times from the same ID_i and the current request is in progress, C_i blocks all these requests from the user until the current session ended. Furthermore, our work can resist *DoS and MITM* attacks[26]. Figure 11 shows the prevention of DoS/ MITM attacks.

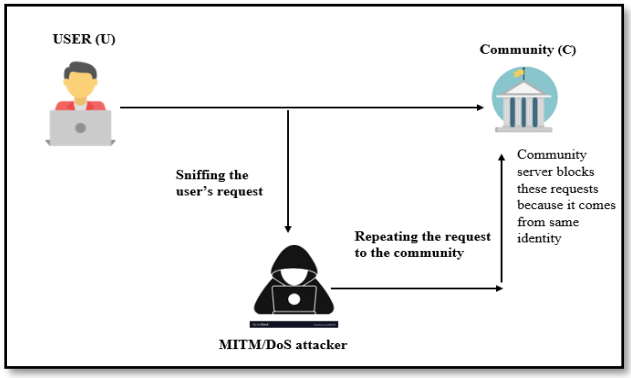


Fig.11: MITM/DoS Attacks prevention

Proposition 9. ***Our proposed scheme secures against insider attack***

Proof. The user's access information in the C_i are ID_i and ID'_i. The attacker tries to use the ID_i for do illegal access to U_i's account. Since the C_i not keep the user's keys, the attacker cannot do illegal access because the authentication server (AS) needs to verify the user's signature based on PK_{Ui}. Therefore, $\bar{A}$ hasn't any information about the keys $(PK_{Ui}, S_{K_{U_i}})$. As result, $\bar{A}$ unable to access/corrupt, or steal the user's sensitive data.

Proposition 10. ***Our proposed scheme secures against device stolen attack***

Proof. We assume the user's device gets stolen; the attacker certainly uses the device from different locations or networks. In the proposed scheme, we solve this problem by embedding a *public IP verification* property to verify the user when he tries to use his account from a different location. When the attacker tries to use the user's account, the proposed scheme takes the current public IP of an attacker (A_{ip}) and compares with the authorized public IPs of the user $(U_{i_{ip}})$. As a result, the attacker was unable to use the user's account from a different network[27]. Table 5 bellows explain the public IP information associated with the user.

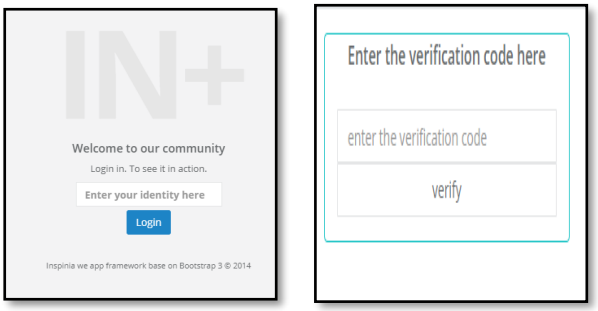
TABLE 5
USER'S AUTHORIZED PUBLIC IPS

PUBLIC IP	User's identity
93.180.220.234	User1
37.237.152.216	User2
37.98.225.192	User1

VI. EXPERIMENTAL RESULTS

1. *Implementation*

To implement and simulate our proposed scheme, we need to install the XAMPP that supports PHP language software on a computer system containing Windows 10 Enterprise operating system (64 bit), Intel (R) Core (TM) i5-4500U CPU @ 2.70 GHz 2.90 GHz processor, and 4 GB RAM. We used the XAMPP software to simulate the community and authentication servers and PHP, MySQL languages for back-end programming and HTML5, JS, jQuery, and Bootstrap for front-end programming. In Fig. 12 (a, b), we explain the implementation of our proposed scheme in a practical way.



a b

Fig.12: The user's interfacing for the verification phase: a. dialog box for enter the identity. b. user's dialog box for enter the verification code.

For the public IP verification property, when the user enters his identity from an unauthorized location, *AS* sends an email to the user containing the verification link as Fig. 13. The AS sets the timestamp value 3 minutes for the validity of the verification link, if the user exceeds this timestamp, the link will be expired as shown in Fig. 13.

Fig.13: The received email from an authentication server

2. *Performance Analysis*

2.1 Computation cost

The computational cost is used to determine the time complexity of the proposed scheme. We compared our work with other related works based on the criteria of [23, 28], where the cost of crypto functions are listed in Table 6. We noticed that the crypto hash function, symmetric and asymmetric encryption/decryption and digital signature is a common operation among the authors. Comparing our scheme with other related works in the term of computational cost is shown in Table 7 and Fig. 14.

TABLE 6
TIME COMPLEXITY FOR CRYPTO OPERATIONS

Authors	Time needed	Result
Our scheme	$4T_h+6T_m$	0.112
***Bojjagani et al* [17]**	$7T_h+6T_m$	0.191
***Lin et al*[21]**	$10T_h+2T_m$	0.24
***Binu et al*[29]**	$9T_h+3T_m$	0.222
***Roy et al*[19]**	$9T_h+1T_m$	0.212
***Lee et al*[24]**	$4T_h+3T_m$	0.107
***Dey et al*[30]**	$5T_h+4T_m$	0.135

TABLE 7
COMPUTATION COST COMPARISION WITH OTHER RELATED WORKS

Term	meaning	Time needed
T_m	Mathematical operation	0.005ms
T_h	Crypto hash function	0.023ms
$T_\oplus$	Exclusive OR operation	negligible

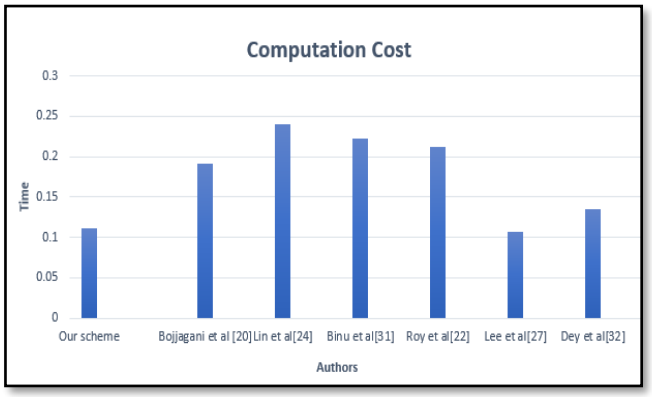


Fig.14: Computation cost comparison

From the above comparisons, the proposed scheme has $(4T_h+6T_m)$= 0.112 has a less time complexity compared with other related works. We note that the proposed scheme owned a good balance between performance and security features (see Table 1).

2.2. Communication cost

To calculate the cost of transmitted messages in the verification phase, we assumed the identity size is 32 bit, the hash value's size is 160 bits, the size of schnorr digital signature is 64 bytes (512 bits)[14, 23], and the size of the ECDSA signature is 72 bytes (576 bits). We also compare our proposed scheme with other related works based on Table 8 below**.**

TABEL 8
COMMUNICATION COST COMPARISON WITH OTHER RELATED WORKS

Authors	No of bits	No of messages
Our scheme	1312	5
***Bojjagani et al.*[17]**	1440	5
***Lin et al.*[21]**	1536	4
***Binu et al.*[29]**	2304	7
***Roy et al.*[19]**	864	2
***Lee et al.*[24]**	1184	7
***Dey et al.*[30]**	1280	4

From the above table, our proposed scheme balanced between security features and the communication bits (see Table 2).

VII. CONCLUSIONS

In this paper, we present a secure verification scheme that avoids famous cyber-attacks like phishing attacks and can detect the existence of an attack. Additionally, our work has several security features like strong verification, forward secrecy, user identity anomaly, and public IP verification property. The *AS* plays pivot rule for verifies the signed message from the user to eliminate the possibility of phishing attacks. The proposed scheme was analyzed formally using the Scyther tool, which confirmed that the proposed scheme has feasible security merits and the results obtained proved that the proposed scheme is safe and secure. We believe that our study and analysis will be helpful to end users, researchers, bankers, mobile app developers, and financial institutions.

CONFLICT OF INTEREST

The authors have no conflict of relevant interest to this article.

REFERENCES

[1] A. Salman, F. Tahir, and M. Rashid, "Design and implementation model for linearization sensor characteristic by FPAA," *Iraq Journal for Electrical and Electronic Engineering*, vol. 11, no. 2, pp.165-173, 2015.

[2] E. Rostami, F. Karlsson, and S. Gao, "Requirements for computerized tools to design information security policies," *Computers & Security*, vol. 99, no.1, pp. 1-17, 2020 .

[3] L. Ma, et al, "Mitigation of malicious attacks on structural balance of signed networks," *Physica A: Statistical Mechanics and its Applications*, vol. 548, p. 123841, 2020.

[4] G. Caporale, W.-Y Kang, F. Spagnolo, and N. Spagnolo, "Cyber-attacks, spillovers and contagion in the cryptocurrency markets," *Journal of International Financial Markets, Institutions and Money*, 2021, in press.

[5] P. Makawana, R. Jhaveri, "A bibliometric analysis of recent research on machine learning for cyber security," *Intelligent communication and computational technologies*, vol. 19, pp.213-226 ,2018.

[6] A. Vishwanath, "Mobile device affordance: Explicating how smartphones influence the outcome of phishing attacks," *Computers in Human Behavior*, vol. 63, pp. 198-207, 2016.

[7] D. Goel, and A. Jain, "Mobile phishing attacks and defence mechanisms: State of art and open research challenges," *Computers & Security*, vol. 73, pp. 519-544, 2018.

[8] S. Curtis, P. Rajivan, D. Jones, and C. Gonzalez, "Phishing attempts among the dark triad: Patterns of attack and vulnerability," *Computers in Human Behavior*, vol. 87, pp. 174-182, 2018.

[9] J. Rastenis, et al, "E-mail-Based Phishing Attack Taxonomy," *Applied Sciences*, vol. 10, no. 7, p. 2363, 2020.

[10] C. Cremers, "The Scyther Tool: Verification, falsification, and analysis of security protocols," In *International conference on computer aided verification*, pp. 414-418. Springer, Berlin, Heidelberg, 2008.

[11] C. Thammarat ,"Efficient and Secure NFC Authentication for Mobile Payment Ensuring Fair Exchange Protocol," *Symmetry*, vol. 12, no. 10, p. 1649, 2020.

[12] C. Schnorr, "Efficient signature generation by smart cards," *Journal of cryptology*, vol. 4, no. 3, pp. 161-174, 1991.

[13] Z. Shao, "Fair exchange protocol of Schnorr signatures with semi-trusted adjudicator," *Computers & Electrical Engineering*, vol. 36, no. 6, pp. 1035-1045, 2010.
[14] J. Katz. and Y. Lindell, "*Introduction to Modern Cryptography (Chapman & Hall/Crc Cryptography and Network Security Series),"*: Chapman & Hall/CRC, 2007.
[15] S. Christopher D. Manning, and P.Raghavan, "*Introduction to information retrieval,"* vol. 39, Cambridge: Cambridge University Press, 2008
[16] C. Huang, S. Ma, W.-L Yeh, C. -Y Lin, and C. -T Lee, "Mitigate web phishing using site signatures," In *TENCON 2010-2010 IEEE Region 10 Conference*, pp. 803-808, IEEE, 2010.
[17] S. Bojjagani, D. Brabin, and P. Rao, "PhishPreventer: A Secure Authentication Protocol for Prevention of Phishing Attacks in Mobile Environment with Formal Verification," *Procedia Computer Science*, vol. 171, pp. 1110-1119, 2020.
[18] D. Johnson, A.Menezes, and S. Vanstone, "The elliptic curve digital signature algorithm (ECDSA)," *International journal of information security*, vol. 1, no. 1, pp. 36-63, 2001.
[19] S.Roy, et al, "On the design of provably secure lightweight remote user authentication scheme for mobile cloud computing services," *IEEE Access*, vol. 5, pp. 25808-25825, 2017.
[20] A. Ahmed-N, and M. Samovar, "Strong authentication for mobile cloud computing," In *2016 13th International Conference on New Technologies for Distributed Systems (NOTERE)*, pp. 1-6, 2016.
[21] H. Lin, "Efficient mobile dynamic ID authentication and key agreement scheme without trusted servers," *International Journal of Communication Systems*, vol. 30, no. 1, p. e2818, 2017.
[22] B. Ross, et al, "Stronger Password Authentication Using Browser Extensions," In *USENIX Security Symposium*, pp. 17-32. 2005.
[23] E. Munivel, and A. Kannammal, "New authentication scheme to secure against the phishing attack in the mobile cloud computing," *Security and Communication Networks*, vol. 2019, pp. 1-19, 2019.
[24] A. Lee, "Authentication scheme for smart learning system in the cloud computing environment," *Journal of Computer Virology and Hacking Techniques*, vol. 11, no. 3, pp. 149-155, 2015.
[25] O. Okunoye, N. Azeez, and F. Ilurimi, "A Web enabled Anti-phishing solution using enhanced Heuristic based technique," vol. 13, no. 2, pp. 304-321, 2017.
[26] K. Mazur, B. Ksiezopolski, and R. Nielek, "Multilevel modeling of distributed denial of service attacks in wireless sensor networks," *Journal of Sensors*, vol. 2016, pp. 1-13, 2016.
[27] P. Waggoner, R. Kennedy, and S. Clifford, "Detecting fraud in online surveys by tracing, scoring, and visualizing IP addresses," *Journal of Open Source Software*, vol. 4, no. 37, pp. 1-5, 2019.
[28] H. Kilinc, and T. Yanik, "A survey of SIP authentication and key agreement schemes," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 2, pp. 1005-1023, 2013.
[29] S. Binu, M. Misbahuddin, and P. Raj,"A strong single sign-on user authentication scheme using mobile token without verifier table for cloud based services," In *Computer and Network Security Essentials*, pp. 237-261, 2018.
[30] S. Dey, S. Sampalli, and Q. Ye, "MDA: message digest-based authentication for mobile cloud computing," *Journal of Cloud Computing*, vol. 5, no. 1, pp. 1-13, 2016.