Journal of Advanced Research in Applied Sciences and Engineering Technology

Journal homepage:https://semarakilmu.com.my/journals/index.php/applied_sciences_eng_tech/index
ISSN: 2462-1943

An Improved Secure Authentication in Lightweight IoT

Noor Afiza Mohd Ariffin[1,*], Vanitha Paliah[1]

[1] Department of Computer Science, Faculty of Computer Science and Technology, Universiti Putra Malaysia, 43400 Serdang, Selangor, Malaysia

ARTICLE INFO

Article history:
Received 16 April 2023
Received in revised form 3 June 2023
Accepted 11 July 2023
Available online 12 August 2023

Keywords:
Authentication; IoT; cloud server

ABSTRACT

Internet of Things (IoT) has been widely accepted by users and with rapid development of cloud computing, users are able to access the IoT services in various environment, including smart home, healthcare and smart factory. However, users are insecure as their data is being transmitted via open communication channel. The previous research protocol does not resist insider attack which leads to insecure scheme. In this proposal, we propose an enhanced security measurement to resist insider attack. The proposed scheme expected to achieve security requirements and resist insider attack. The proposed scheme will be validated using the automated validation of internet security protocols and applications (AVISPA) simulation tool and we will compare the performance and security features of the existing scheme and the proposed scheme.

1. Introduction

Information technology has evolved tremendously over the years and innovated new technologies to improve our lives. Therefore, wireless access technology has been improved to accommodate our need. We are in third industrial revolution called as Digital Age where we have moved from analogic to digital process. As we move forward, researchers are innovating new communication technologies such Wi-Fi, 4G, 5G, LTE, RFID and others which being offered to users to be able to utilize in day to day usage using devices such as tablets and smartphones [4].

Internet has changed our live style as well as the way we are working. Businesses are emerging through artificial intelligence (AI), Big Data and Cloud Computing and the Internet of Things (IoT). These technologies have simplified our lives which efficiently reduce human effort and allow users to communicate around the world and share data instantly [11].

Scientist Ashton has introduced Internet of Things (IoT) in the year of 1999 where sensor devices, smart objects and software are interconnected through network. The idea of IoT is to be communicated anytime and anywhere while collecting data and share it. As we are in 21st Century, IoT has become more popular. Almost every smart device is connected to each other and it is being widely accepted by industrial such as home automation, industrial automation, medical aids, mobile

* *Corresponding author.*
E-mail address: noorafiza@upm.edu.my

https://doi.org/10.37934/araset.31.3.191207

health-care, elderly assistance, intelligent energy management and smart grids, automotive, traffic management, and many others [4].

A survey has been done by El-Hajj *et al*., [4] and they have classified IoT applications into following categories: (a) Internet of sensors (IoS), (b) Internet of energy (IoE), (c) machine to machine (M2M) and (d) Internet of Vehicles (IoV) as shown in Figure 1 [5].

Cloud computing offered four types of deployment models. First, public cloud is a platform where services are available for user in open communication channel like Google Drive or iCloud. Second, private cloud where services are provided through secure communication channel to the users. Third, community cloud where the cloud platform is shared by several organizations. Fourth, hybrid cloud where services are provided in a platform where public and private share the same resources. Data used in IoT devices are being stored in cloud server which used an open communication channel to travel through during the process of fetching and storing data. This represents the security disadvantage as it is possible for the devices to collect sensitive and personal data [4].

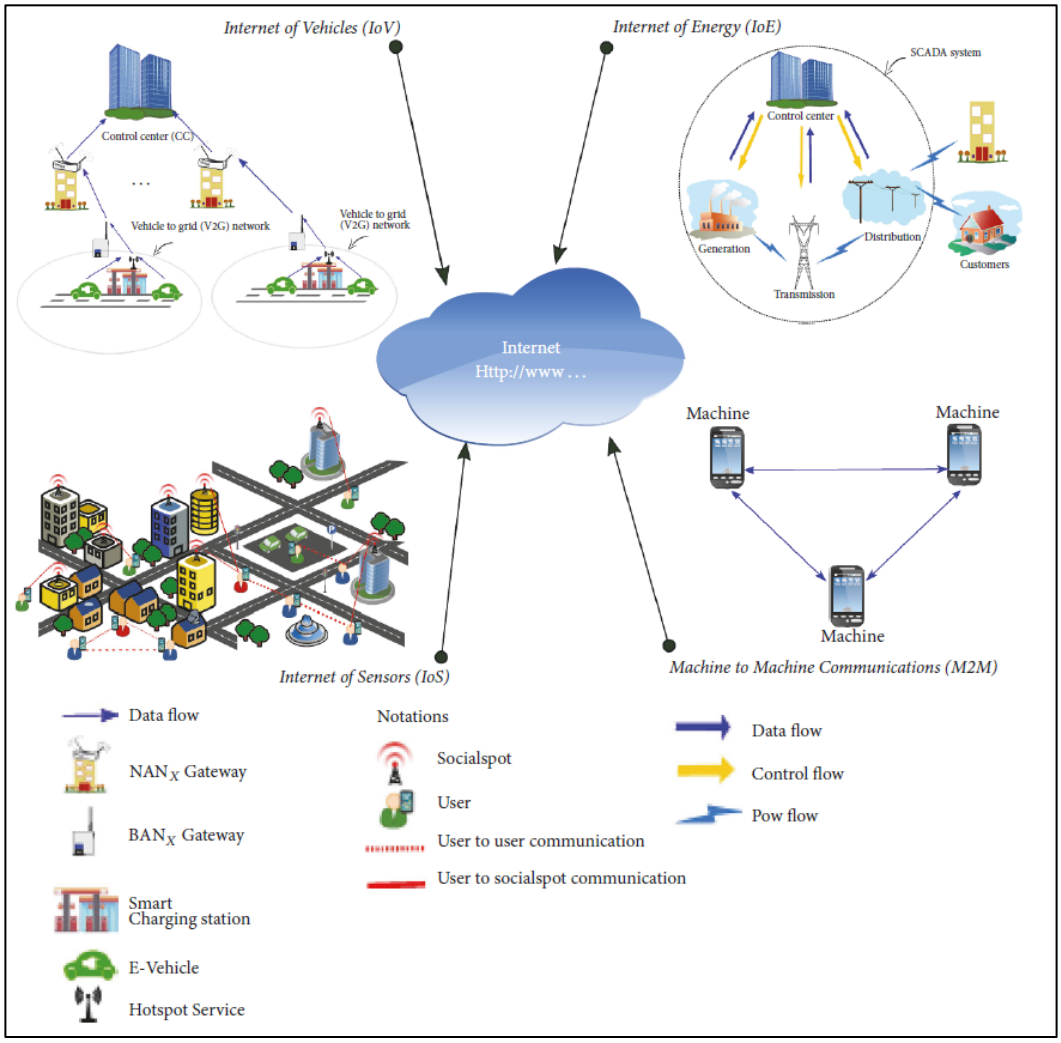


Fig. 1. IoT Categories

2. Research Background

As an improvement from the previous work, multifactor authentications using Duo application are proposed to solve the security issues of an unauthorised access in cloud system. Duo application authentication methods provides variety of ways for user to securely and quickly log in. Duo Push, sent by Duo Mobile authentication app, allows users to approve push notifications to verify their identity. Duo also support Universal 2nd Factor (U2F) security tokens, hardware tokens, mobile passcodes, SMS, phone call back and biometrics like Touch ID to provide flexible and accessible options for all types of users. Duo's single sign-on (SSO) integrates security with the ease of logging in just once to access multiple cloud applications, allows consolidating authentication and identity solutions into one. Duo is designed with asymmetric cryptography to sign and verify communications between Duo's servers and a user's smartphone. A private key stay on the mobile device, and is used to sign all authentication responses, while the public key is used to verify the signature on the server side. That means an attacker cannot access accounts even if they breached Duo's servers. Figure 2 clearly show a duo secure design.

By default, Duo will present three methods to authenticate. Firstly, user must enter their credential, which is username and password. If the username is incorrect, it will disable them to proceed to the next page for password input. If the username is correct, the user must enter their password. Then, they must enter the passcode. User can select options on how to get the passcode. These options include "duo push" where user needs to install the mobile application in order to get the passcode. Second option is "text me" where passcode will be delivered through text to the user. Last option is "call me" which the user will get the passcode through the call. The proposed solution as shown below in Figure 3.

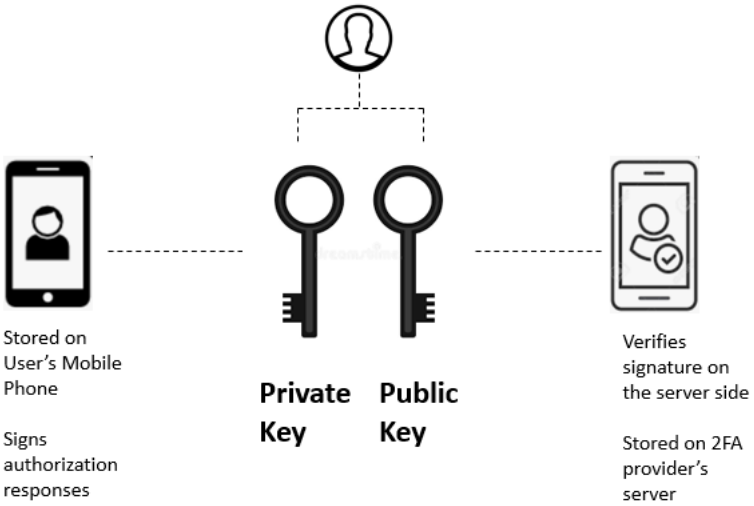


Fig. 2. Duo Secure Design

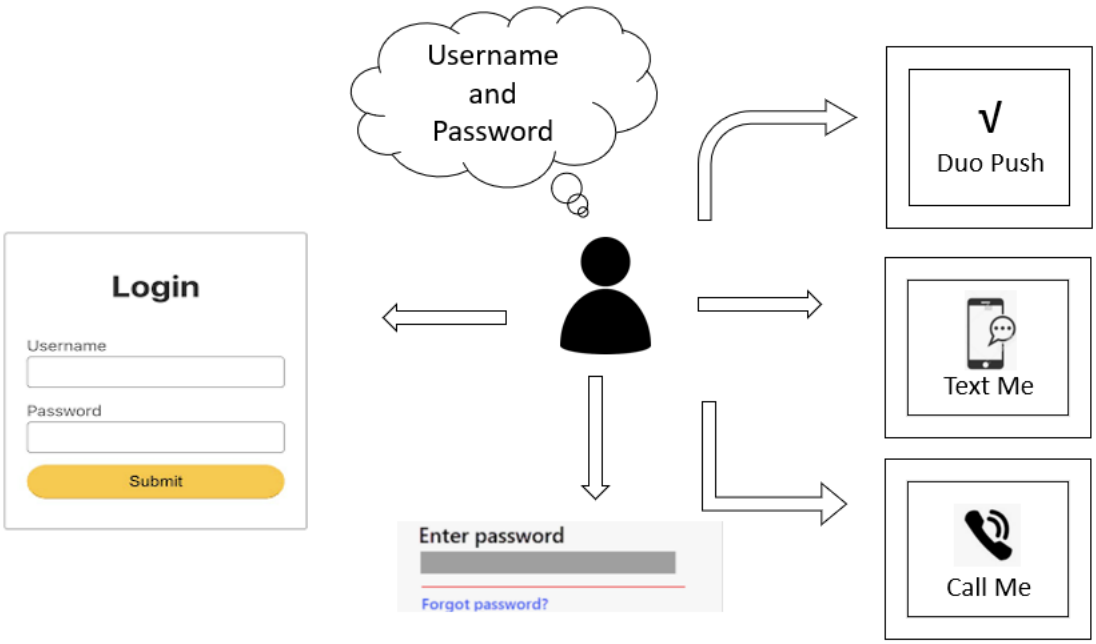


Fig. 3. Proposed Solution

2.1 Authentication

The server will determine whether the user allowed to access the system or not. In this phase, there are three options by default that are given to the user. These options include duo push, text me and call me. The admin can set for Biometric authentication as Touch ID and Face ID.

After completing Duo enrolment, when user try to perform a browser-based login to a web service or application protected with Duo, the Duo authentication will be prompted. Duo can be configured as how the policy is required by the organization. As for the security measurement, Duo Mobile's Security Check up verifies device settings against Duo's recommended security settings and will let user know the device's settings if does not match. Administrator can choose to block access to applications from devices not managed by the organization. If this policy is enforced, then users will not be able to complete Duo authentication from their personal device

2.2 Two Factor Authentication

Two factor authentications have been developed as an initiative to enhance the security measurement between multiple devices. In 2012, Kumar *et al*., [8] proposed two factors authentication with key agreement (AKA) method. While in 2013, Shi *et al*., [12] proposed an AKA protocol using elliptic curve cryptography and Li *et al*.,[27] used a dynamic identity–based AKA scheme. However, in 2015, Wu *et al*., [13] analysed the scheme used in previous research and stated that it could not defend against user impersonation attack and sensor node capture attack. Amin *et al*., [1] proposed a network structure for patient monitoring healthcare system and gave a mutual authentication scheme to achieve user's anonymity. But in 2017, Jiang *et al*., [7] has analysed Amin's study and proved that it fails to resist device stolen attack and desynchronization attack.

2.3 Mutual Authentication

In 2016, Ferrag *et al*., [5] has proposed a guarantee the entity mutual authentication and secure key agreement based on 3GPP standard with three domains network modal, including access networks, evolved packet core, and non-3GPP domain. However, their study does not consider storage cost and the overhead computational cost a higher [5]. This paper describes a few matching methodologies and their limitations in previous research. In Table 1 below, a few researchers matching methodologies and limitations are listed.

Table 1
Comparison with previous research

Title	Author	Methodology	Limitations
A remote password authentication scheme for multiserver architecture using neural networks, IEEE Trans. Neural Network (2001)	L. Li, I. Lin, M.S. Hwang, [28]	Using neural network by identifying the pattern	Heavy time computation, not suitable for IoT
An efficient and practical solution to remote authentication: Smart card. (2002)	Chien, H.Y., Jan, J. and Tseng, Y.M. [29]	Two factor authentication scheme using password and smart cards.	Vulnerable to smart card stolen attack
An efficient multi-server password authenticated key agreement scheme using smart cards with access control (2004)	Chin-Chen Chang, and Jui-Yi Kuo [30]	key agreement scheme basedon the hash function and symmetric key cryptosystem	Lack of efficiency and vulnerable to smart card stolen attack
A lightweight authentication protocol for IoT-enabled devices in distributed cloud computing environment. (2016)	Ruhul Amin, Neeraj Kumar, Rahat Iqbal, Victor Chang [31]	Used mutual authentication and key agreement for authentication	Vulnerable to privileged-insider and impersonation attacks
An efficient authentication and key agreement scheme for multi-gateway wireless sensor networks in IoT deployment (2017)	Fan Wu, Lili Xu, Saru Kumari, Xiong Li, Jian Shen, Kim Kwang Raymond Choo, Mohammad Wazid and Ashok Kumar Das [32]	Proposed three factor authentication and key agreement scheme	Vulnerable to user forgery attack
A secure dynamic ID based remote user authentication scheme for multi-server environment (2018)	Yi-Pin Liao and Shuenn Shyang Wang [33]	Only uses hashing functions to implement a robust authentication scheme	Vulnerable to an insider attack, masquerade attack, server spoofing attack, and registration centre spoofing attacks

Lightweight IoT-based authentication scheme in cloud computing circumstance (2018)	Lu Zhoua, Xiong Li, Kuo-Hui Yeh, Chunhua Su, Wayne Chiu [34]	Hash function and Exclusive –OR operation for encryption	Failed to protect user ID in cloud server

3. Methodology

This section presents the research methodology which included an improved research framework proposed for IoT in cloud computing environment that will informatively be discussed in this study. Table 2 below are the notations used in this study.

Table 2
Notation used

Symbol	Description
IDcs,x	the identity and secret key of CS
Sj,SIDj,PSIDj	the j−th cloud server with its identity and pseudo-identity
Ui, IDi,PIDi,PWi	the i−th user with his identity, pseudo-identity and password
A	the adversary
h(·)	hash function
$\oplus$	exclusive-or function
||	concatenation function
SKu,SKs,SKcs	session keys for Ui, Sj and CS
M1, M2, M3, M4	messages in the authentication

3.1 Research Framework

By adapting the authentication scheme method implemented by Zhou *et al*., [15], three phases of authentication scheme will be used. The proposed scheme includes the improvement on mutual authentication to provide stronger security in cloud computing environment.

As for the network architecture, we will have three entity. First will be the user who will be registered by control server (CS). User will be equipped with device. In this study we will adapt the method used by Zhou *et al*., [15] where user will be issued a smart card in their smart phone. Second is cloud server which will be used by user. There will be many cloud servers which is distributed among themselves and the nearest cloud server will be used by user.

Third is control server which is responsible for user and server registration, authentication and to store user identity and server information. Additionally, if user wants to login to specific server, authentication will be processed and validate by CS. Figure 4 below show the proposed framework in this research.

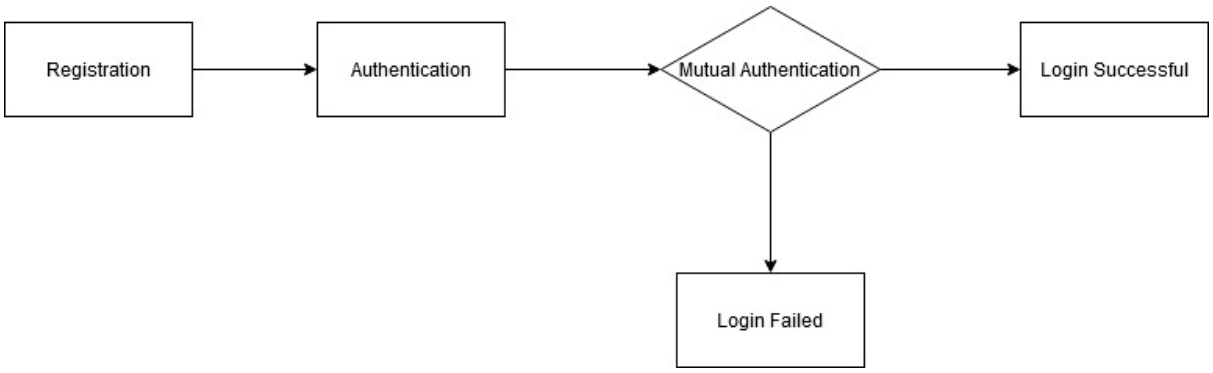


Fig. 4. Proposed Framework

3.2 Registration Phase

Registration phase will be divided in user registration and cloud server registration sub-phases. It will be process through control server (CS).

3.2.1 User Registration Sub-Phase

In user registration sub-phase, the user (U*i*) will be registered by Control Server (CS).

- Step 1: U_i selects its own identity and pseudo-identity pair (ID_i,PID_i), with his own password PW_i and a nonce b_i. He calculates $HP_i = h(PW_i||b_i)$ and sends (ID_i,PID_i) to *CS* via the secure channel.
- Step 2: *CS* checks ID_i first and if it is invalid, the registration will be stopped. If passed, *CS* calculates $C_1* = h(PID_i||ID_{cs}||x)$ and $C_2* = h(ID_i||x)$, stores ID_i in database, and sends (C_1*, C_2*, *IDcs*) to U_i via the secure channel.
- Step 3: U_i computes $C_1 = C_1^* \oplus HP_i$, C and $C_3 = b_i \oplus h(ID_i||PW_i)$ and stores (C_1, C_2, C_3, $PIDi$,ID_{cs}) in his own smart card$_2 = C_2^* \oplus h(ID_i||HP_i)$.

3.2.2 Cloud Server Registration Sub-Phase

In server registration sub-phase, server (Si) will also be registered by CS and stored the information in CS.

- Step 1: The server S_j sends its identity and pseudo-identity pair (SID_j,$PSID_j$) to *CS* via a secure channel.
- Step 2: *CS* computes $B_1 = h(PSID_j||ID_{cs}||x)$ and $B_2 = h(SID_j||x)$, stores SID_j and sends (B_1, B_2, *IDcs*) to S_j via the secure way.
- Step 3: S_j stores (B_1, B_2, *SIDj*, $PSID_j$, ID_{cs})

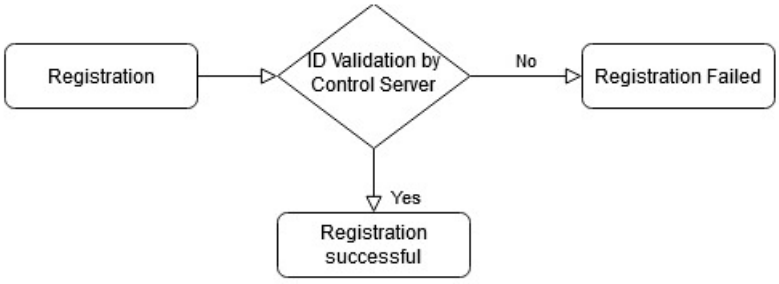


Fig. 5. Registration Phase

3.3 Authentication Phase

There are five steps in this phase.

- Step 1: When U_i wants to access the service of cloud server, he inserts the smart card and enters (ID_i,PW_i). Then the smart card selects a random number r_u and a new pseudo-identity $PID^{new}{}_i$, and computes the following data: $b_i = C_3 \oplus h(ID_i||PW_i)$, $HP_i = h(PW_i||b_i)$, $C_1^* = C_1 \oplus HP_i$, $C_2^* = C2 \oplus h(IDi||HPi)$, $D1 = C_1* \oplus ru$, $D2 = h(ru||PIDi||IDcs) \oplus IDi$, $D3 = C_2* \oplus h(ID_i||HP_i) \oplus PID^{new}{}_i \oplus h(r_u||ID_i)$, $D_4 = h(ID_i||PID_i||PID^{new}{}_i||r_u||D_3)$. Then the message $M_1 = \{PID_i, D_1, D_2, D_3, D_4\}$ is sent to the nearest cloud server S_j.

- Step 2: S_j selects a new pseudo-identity $PSID^{new}{}_j$ and a random number r_s, computes $D_5 = B_1 \oplus r_s$, $D_6 = h(r_s||PSID_j||ID_{cs}) \oplus SID_j$, $D_7 = B_2 \oplus PSID^{new}{}_j \oplus h(r_s||SID_j)$ and $D_8 = h(SID_j||PSID_j||PSID^{new}{}_j||r_s||D_7)$. Then the message $M_2 = \{PIDi, D_1, D_2, D_3, D_4, PSID_j, D_5, D_6, D_7, D_8\}$ is sent to *CS*.
- Step 3: *CS* computes $r_u = D_1 \oplus h(PID_i||ID_{cs}||x)$, and $ID_i = D_2 \oplus h(r_u||PID_i||ID_{cs})$, $PID^{new}{}_i = D_3 \oplus h(ID_i||x) \oplus h(r_u||ID_i)$, and checks if ID_i is valid and D_4? = $h(ID_i||PID_i||PID^{new}{}_i||r_u||D_3)$. Then it continues to compute $r_s = D_5 \oplus h(PSID_j||ID_{cs}||x)$, $SID_j = D_6 \oplus h(r_s||PSID_j||ID_{cs})$, $PSID^{new}{}_j = D_7 \oplus h(SID_j||x) \oplus h(r_s||SID_j)$ and checks if SID_j is valid and D_8? = $h(SID_j||PSID_j||PSID^{new}{}_j||r_s||D_7)$. If any verification is wrong, the session will be stopped. Otherwise, *CS* selects a random number r_s, and calculates $SK_{cs} = h(r_u \oplus r_s \oplus r_{cs})$, $D_9 = h(PSID^{new}{}_j||ID_{cs}||x) \oplus h(r_s||PSID^{new}{}_j)$, $D10 = h(PSID^{new}{}_j||rs||PSIDj) \oplus (ru \oplus rcs)$, $D11 = h(SKcs||D9||D10||h(SIDj||x))$, $D_{12} = h(PID^{new}{}_i||ID_{cs}||x) \oplus h(r_u||PID_i^{new})$, $D_{13} = h(PID^{new}{}_i||r_u||PID_i) \oplus (r_s \oplus r_{cs})$ and $D_{14} = h(SK_{cs}||D_{12}||D_{13}||h(ID_i||x))$. In this phase, control server will add another server response challenge to authenticate server and user mutually. Control server, the message $M_3 = \{D9, D10, D11, D12, D13, D14\}$ is sent to *Sj*.

- Step 4: S_j computes $(r_u \oplus r_{cs}) = D_{10} \oplus h(PSID^{new}{}_j||r_s||PSID_j)$, and $SK_s = h(r_s \oplus r_u \oplus r_{cs})$, and checks if D_{11}? = $h(SK_s||D_9||D_{10}||B_2)$ is correct. If so, S_j calculates $B^{new}{}_1 = D_9 \oplus h(r_s||PSID^{new}{}_j)$ and replaces (B_1,$PSID_j$) with ($B^{new}{}_1$,$PSID^{new}{}_j$) and ensure the nonce number is correct to authentication with user. Then message $M_4 = \{D_{12}, D_{13}, D_{14}\}$ is sent to U_i.

- Step 5: The smart card computes $(r_s \oplus r_{cs}) = D_{13} \oplus h(PID_i^{new}||r_u||PID_i)$, and $SK_u = h(r_u \oplus r_s \oplus r_{cs})$, and checks D_{14}? = $h(SK_u||D_{12}||D_{13}||C_2*)$ and if the nonce number is same as sent by server. If so, the card computes $C_1{}^{new} = D_{12} \oplus h(r_u||PID^{new}{}_i) \oplus HP_i$ and replaces (C_1,PID_i) with ($^{C}{}_1new, PIDnew_i$). When nonce number are the same by server and user, it has achieved the mutual authentication. If failed, session will be terminated.

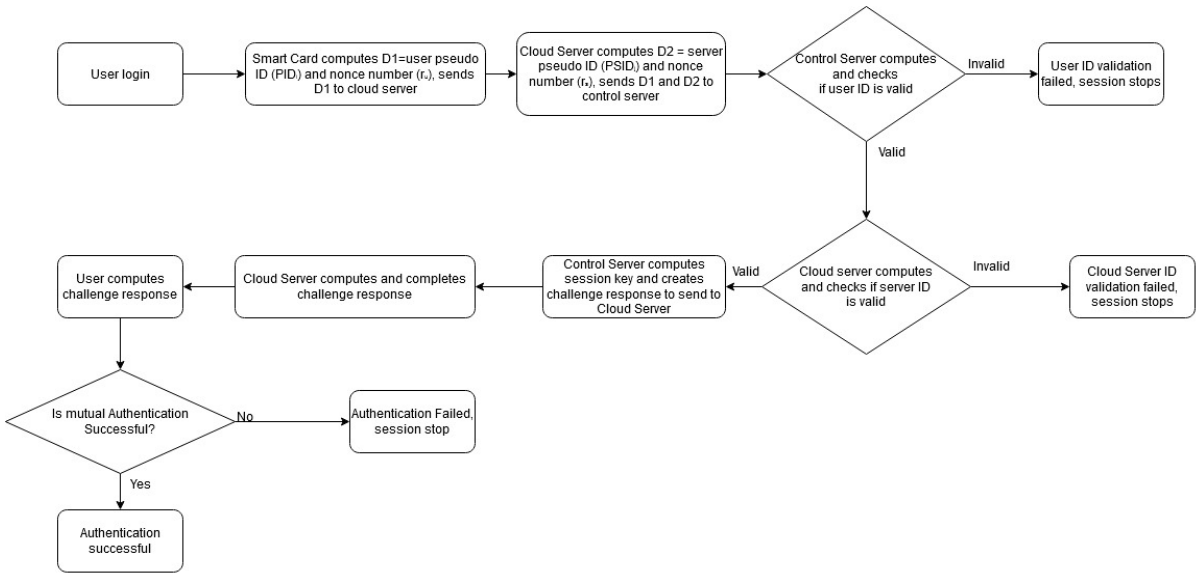


Fig. 6. Authentication Phase

3.4 Password Change

There are several steps involve in the Password Change.

- Step 1: If U_i wants to modify his own password, he first makes operations as the step 1 in Section 3.2, but a password change request is accompanying with the message M_5 which is sent to *CS* and $M_5 = M_1$.

- Step 2: *CS* computes r_u, ID_i and PID_i, and checks ID_i and D_4. If passed, *CS* calculates D_{12} and $D_{15} = h(ID_i||PID_i||PID^{new}_i||r_u||D_{12})$. Finally, it sends $M_6 = \{D_{12}, D_{15}\}$ with a permission.

- Step 3: The smart card checks $D_{15}? = h(ID_i||PID_i||PID_i^{new}||r_u||D_{12})$. If so, it asks U_i to input PW_i^{new} as a newpassword,computes $HPinew = h(PWinew||bi)$, $C_1new2 = D12 \oplus h(ru||PIDinew) \oplus HPinew$, $C_2new = C2* \oplus h(ID_i||HP_i^{new})$ and $C_3^{new} = b_i \oplus h(ID_i||PW_i^{new})$, and replaces (C_1, C_2, C_3, PID_i) with $(^{C}_1new2, ^{C}_2new, ^{C}_3new, PIDnew_i)$.

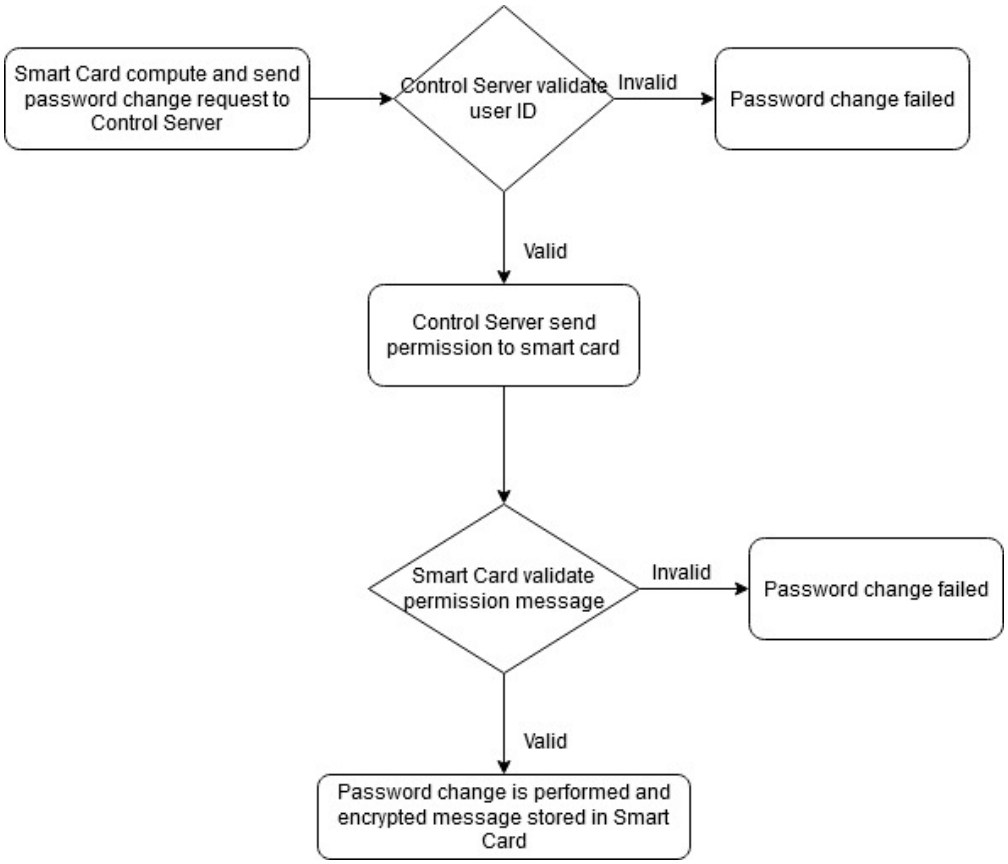


Fig. 7. Password Change Phase

4. Implementation

4.1 Overview

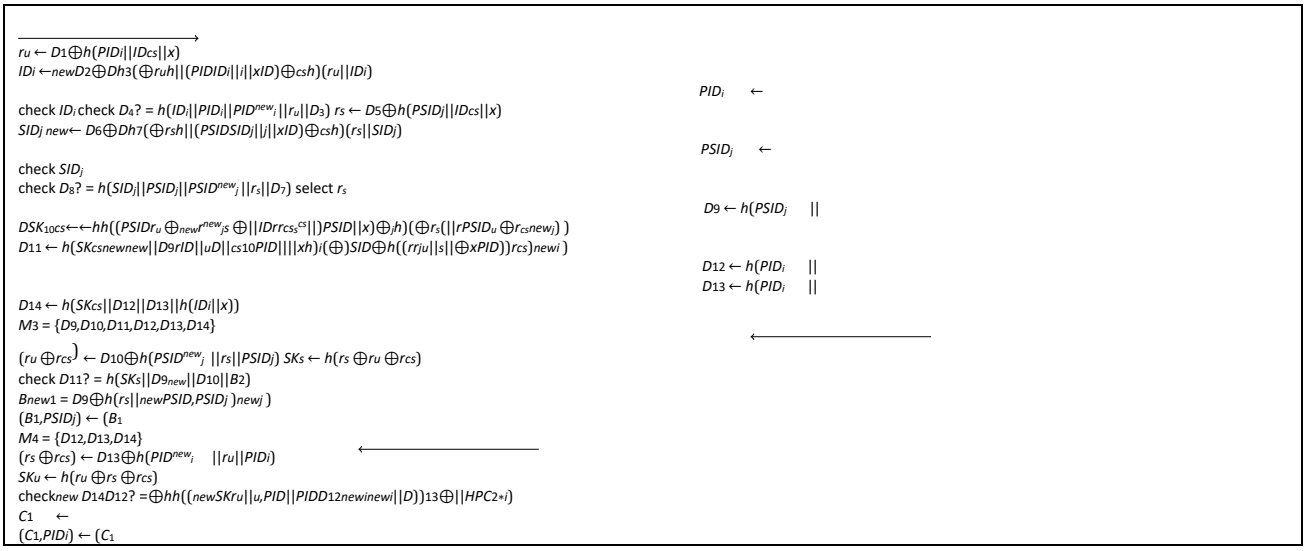
The implementation of proposed framework will be discussed in this chapter along with tools and techniques used.

Proverif tool is used to make formal verification of proposed framework using cryptography protocol. Blanchet *et al*.,[35] has documented process of Proverif and until today it has been used to make formal verification of security protocol and widely accepted.

Table 3 is the authentication process adapted from Zhou *et al*., [15] proposed scheme and has been improvised by adding another entity for mutual authentication.

Table 3
Authentication process

U i S j CS input ID_i, PW_i select $ru, PID^{new}_i, bi = C3 \oplus h(IDi \| PWi)$ $HPi \leftarrow h(PWi \| bi)$ $C1* \leftarrow C1 \oplus HPi$
$C2* \leftarrow = C2 \oplus h(IDi \| HPi)$ $D1 \leftarrow C1* \oplus ru$
$D2 \leftarrow h((rIDu \| i \| PIDi \| \| i \| IDPIDcs) \oplus newi \oplus ID \| riunew \| D3 \oplus) h(ru \| IDi)$
$D3 \leftarrow C2* \oplus h(IDi\ HPi)$ $PIDi$
$D4 \leftarrow h$ PID
$M1 = \{PIDi, D1, D2, D3, D4\}$
$\longrightarrow$ select $PSID^{new}_j$, rs
$D5 \leftarrow B1 \oplus rs$
$D6 \leftarrow h(rs \| \| PSIDnewj \| \| IDcsh)(r \oplus s \| \| SIDnewSIDjjr)s \| \| D7)$
$D7 \leftarrow B2 \oplus PSID_j \oplus$
$D8 \leftarrow h(SIDj \| \| PSIDj \| \| PSID_j$ ||
$M2 = \{PIDi, D1, D2, D3, D4, PSIDj, D5, D6, D7, D8\}$



4.2 Code

Figure 8 shows the identifier of the process. x is the secret key of control server, IDi and PWi are the user identity and password of the user. sch1 and sch2 identified as secure channels and ch1 and ch2 identified as public channels. SKu, SKs and SKcs are session keys dedicated for user, cloud server and the control server. SIDj and IDcs are identities of the cloud server and the control server. Database, D1 stores the user's identity information and D2 stores the cloud servers identity information. h, xor and con are symbols for hash function, exclusive-or operation and concatenation operations. Then, two pairs of events are defined as the starts and ends of user and cloud server authentication process. Finally, five queries referring three session keys and the orders of the two pairs of events are listed.

```
free x:bitstring [private].
free IDi:bitstring [private].
free PWi:bitstring [private].
free sch1: channel [private].
free sch2: channel [private].
free SKu: bitstring [private].
free SKs: bitstring [private].
free SKcs: bitstring [private].
free ch1: channel.
free ch2: channel.
const SIDj:bitstring.
const IDcs:bitstring.
table D1(bitstring).
table D2(bitstring).
fun h(bitstring):bitstring.
```

```
fun xor(bitstring,bitstring):bitstring.
fun con(bitstring,bitstring):bitstring.
equation forall m:bitstring,n:bitstring;
xor(xor(m,n),n)=m.
event UStart(bitstring).
event UAuth(bitstring).
event SStart(bitstring).
event SAuth(bitstring).
query attacker(SKu).
query attacker(SKs).
query attacker(SKcs).
query id:bitstring; inj-event(SAuth(id))
==> inj-event(SStart(id)).
query id:bitstring; inj-event(UAuth(id))
==> inj-event(UStart(id)).
```

Fig. 8. Identifier

The code of user and cloud server is coded in Figure 9. Left table showed the registration part of user is the code from line two to nine and followed by authentication process. Similarly, in the right

table, registration of cloud server is coded from line two to six and followed by authentication process.

In Figure 10, the code for control server is coded. In the left table, the code correspondent to user registration and cloud server registration is separated by two dot-lines. We use UReg and SReg to denote the two parts. The rest part, except the last line in the right rectangle, are the authentication part for cloud server. The last line let CS =URegk SRegk CSAuth. expresses that the process of control server is constructed by the three parts: UReg, SReg and CSAuth.

The results of the queries are shown in Figure 11. As shown in the result, two pairs of events are stable, and the session keys are robust against simulated attackers. So, the scheme is validated as secure via the formal verification using Proverif. Analysis will be discussed in next chapter.

```
let U=
new bi:bitstring; new PIDi:bitstring; let
HPi=h(con(PWi,bi)) in out(sch1,(IDi,PIDi));
in(sch1,(uC1reg:bitstring,uC2reg:bitstring
)); let C1 = xor(uC1reg,HPi) in let C2 =
xor(uC2reg,h(con(IDi,PWi))) in let C3 =
xor(bi,h(con(IDi,PWi))) in
!
(
event UStart(IDi); new ru:bitstring; new
PIDinew:bitstring; let ubi=
xor(C3,h(con(IDi,PWi))) in let uHPi =
h(con(PWi,ubi)) in let C1bar =
xor(C1,uHPi) in let C2bar =
xor(C2,h(con(IDi,uHPi))) in let D1 =
xor(C1bar,ru) in let D2 =
xor(h(con(con(ru,PIDi),IDcs)),IDi) in let D3
= xor(xor(xor(C2bar,h(con(IDi,uHPi))),
PIDinew),h(con(ru,IDi))) in
let D4 =
h(con(con(con(con(IDi,PIDi),PIDinew),
ru),D3)) in
let M1 = (PIDi,D1,D2,D3,D4) in
out(ch1,M1);
in (ch1,(uD12:bitstring,uD13:bitstring,
uD14:bitstring));
let SCS =
xor(uD13,h(con(con(PIDinew,ru), PIDi)))
in
let SKu = h(xor(SCS,ru)) in
if uD14 =
h(con(con(con(SKu,uD12),uD13), C2bar))
then
let C1new =
xor(xor(uD12,h(con(ru,PIDinew))), uHPi)
```

```
let S =
new PSIDj:bitstring; out(sch2,(SIDj,PSIDj));
in(sch2, (sB1reg:bitstring,sB2reg:bitstring)); let
B1 = sB1reg in let B2 = sB2reg in
! (
in(ch1,(sPIDi:bitstring,sD1:bitstring,
sD2:bitstring,sD3:bitstring,sD4:bitstring));
event SStart(SIDj); new rs:bitstring; new
PSIDjnew:bitstring; let D5 = xor(B1,rs) in
let D6 = xor(h(con(con(rs,PSIDj),IDcs)), SIDj) in
let D7 = xor(xor(B2,PSIDjnew),h(con(rs, SIDj)))
in
let D8 = h(con(con(con(con(SIDj,PSIDj),
PSIDjnew),rs),D7)) in
let M2 = (sPIDi,sD1,sD2,sD3,sD4,PSIDj,
D5,D6,D7,D8) in out (ch2,M2);
in(ch2,(sD9:bitstring,sD10:bitstring,sD11:bitstri
ng,
sD12:bitstring,sD13:bitstring,sD14:bitstring));
let UCS = xor(sD10,h(con(con(PSIDjnew,
rs),PSIDj))) in
let SKs = h(xor(UCS,rs)) in if sD14 =
h(con(con(con(SKs,sD9), sD10),B2)) then
let B1new = xor(sD9,h(con(rs, PSIDjnew))) in let
PSIDj = PSIDjnew in let B1 = B1new in let M4 =
(sD12,sD13,sD14) in
out(ch1,M4);
0
).
```

```
in let C1 = C1new in let PIDi = PIDinew in
0
).
```

Fig. 9. Registration and Authentication

```
let UReg =
in(sch1,(csIDireg:bitstring,csPIDireg:bitstring
)); let csC1reg =
h(con(con(csPIDireg,IDcs),x)) in let csC2reg
=h(con(csIDireg,x)) in insert D1(csIDireg);
out (sch1,(csC1reg,csC2reg)).

_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
let SReg =
in(sch2,(csSIDjreg:bitstring,csPSIDjreg:bitstri
ng)); insert D2(csSIDjreg);
let csB1reg = h(con(con(csPSIDjreg,IDcs),x))
in let csB2reg = h(con(SIDj,x)) in
out(sch2,(csB1reg,csB2reg)).

_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
let CSAuth =
in(ch2,(csPIDi:bitstring,csD1:bitstring,
csD2:bitstring,csD3:bitstring,csD4:bitstring,
csPSIDj:bitstring,csD5:bitstring,csD6:bitstrin
g, csD7:bitstring,csD8:bitstring)); new
rcs:bitstring; let csru = xor(csD1,
h(con(con(csPIDi,IDcs),x)) )in let csIDi =
xor(csD2,h(con(con(csru,csPIDi), IDcs))) in
get D1(=csIDi) in
let csPIDinew =
xor(xor(csD3,h(con(csIDi,x))),
h(con(csru,csIDi))) in
if csD4 = h(con(con(con(con(csIDi,csPIDi),
csPIDinew),csru),csD3)) then
event UAuth(csIDi);
let csrs =
xor(csD5,h(con(con(csPSIDj,IDcs
), x))) in let csSIDj =
xor(csD6,h(con(con(csrs,csPSIDj
), IDcs))) in get D2(=csSIDj) in
let csPSIDjnew =
xor(xor(csD7,h(con(csSIDj,
x))),h(con(csrs,csSIDj))) in if
csD8 =
h(con(con(con(con(csSIDj,csPSI
Dj), csPSIDjnew),csrs),csD7))
then event SAuth(csSIDj);
let SKcs =
h(xor(xor(csru,csrs),rcs)) in let
D9 =
xor(h(con(con(csPSIDjnew,IDcs),
x)), h(con(csrs,csPSIDjnew))) in
let D10 =
xor(h(con(con(csPSIDjnew,csrs),
csPSIDj)),xor(csru,rcs)) in
let D11 =
h(con(con(con(SKcs,D9),D10),
h(con(csSIDj,x)))) in
let D12 =
xor(h(con(con(csPSIDjnew,IDcs),
x)),h(con(csru,csPIDinew))) in
let D13 =
xor(h(con(con(csPIDinew,csru),
csPIDi)),xor(csrs,rcs)) in
let D14 =
h(con(con(con(SKcs,D12),D13),
h(con(csIDi,x)))) in let M3
=(D9,D10,D11,D12,D13,D14) in
out(ch2,M3).
let CS = UReg | SReg | CSAuth.
```

Fig. 10. Control Server Process

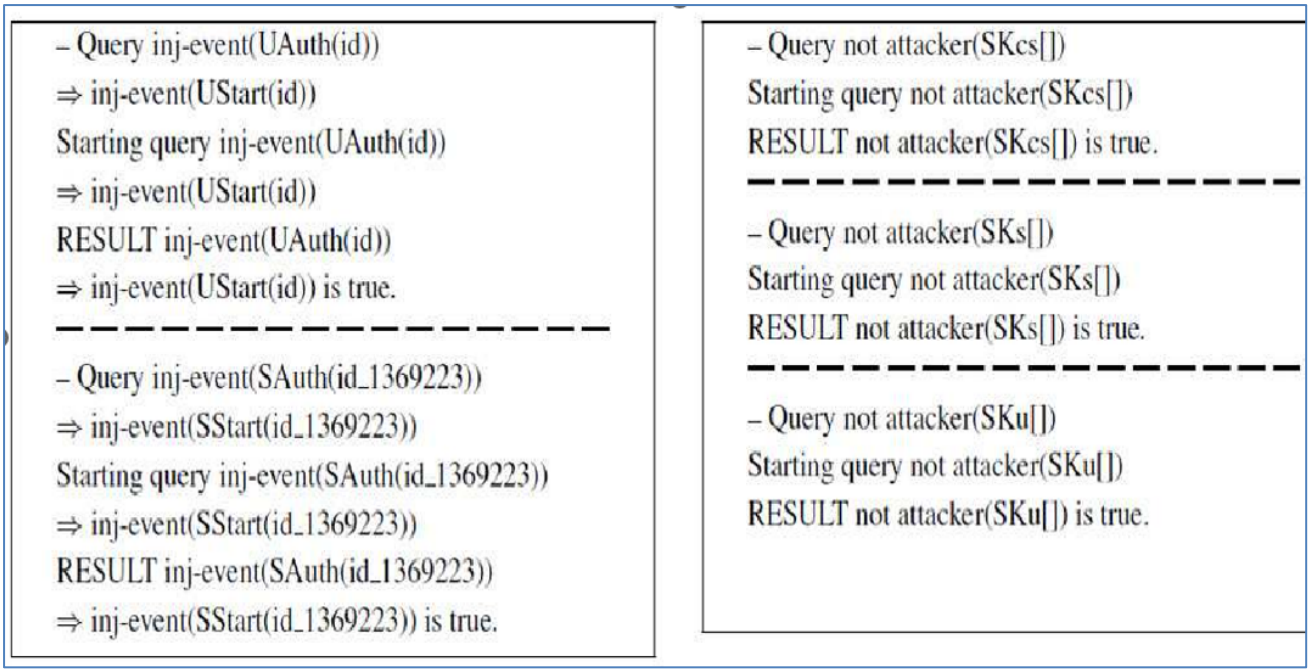


Fig. 11. Result

5. Results

This section discusses about the general security requirements of a robust authentication scheme for IoT and cloud servers. Then, a comparison table, Table 5 which shows the requirement met by our proposed scheme and adapted schemes from Zhou *et al*., [15] is demonstrated.

Table 4
Result

Attack type	Zhou *et al*., [15]	Proposed Scheme
Resistance to Insider Attack	No	Yes
Mutual Authentication	No	Yes

Table 5
Computing Cost

Computational Cost	Zhou *et al*., [15]	Proposed Scheme
Time cost for Ui (ms)	10Th = 0.0517	10Th = 0.0527
Time cost for SJ (ms)	7Th = 0.0362	7Th = 0.0369
Time cost for CS (ms)	19Th = 0.0983	19Th = 0.1002
Communication cost (bits)	5856	6332

Accordingly to the table above, our proposed scheme achieves the security requirements and objective of this study while Zhou *et al*., [15] claimed that their scheme achieved mutual authentication. However, as discussed, server and user did not authentication each other to achieve mutual authentication and this can be improvised. User's authentication or password change is

recorded in database audit trails, this will help forensic investigation as well where all the information will be stored database. Moreover, in our scheme, D4 and D8 are verified by control server so that user and server is authenticated by control server. Control server then will send challenge response to cloud server where both user and server must authenticate each other to achieve mutual authentication. If the challenge response failed, authentication will be failed too. Hence, mutual authentication is achieved.

Although computational cost has been increased, we believe security has been enhanced and additional process has increased the computation cost. Considering the security layer that has been enhanced, we argue the proposed scheme performance are acceptable.

6. Conclusions

We have discussed the importance and requirement of security and efficiency requirements in the previous schemes which was designed for cloud computing environment with IoT-enable devices. Then, we introduce an improvised secure scheme for IoT-cloud architecture circumstances. We have concluded our analyse with a formal verification analysis provided by Proverif and the security discussion. Our proposed authentication scheme is proven to be secure against insider attacks and achieves mutual authentication at the same time. The performance is a little high then the compared protocol as we have added additional process but consider the security of the protocol, we argue that our proposed authentication scheme is highly suitable for real IoT-cloud circumstances in real world due to satisfactory for security and practicality requirements.

Acknowledgement

This research was supported by Geran Putra IPM Project Vote Number: 9679500, Universiti Putra Malaysia (UPM). Special thanks from authors for financial support from Universiti Putra Malaysia.

References

[1] Amin, Ruhul, SK Hafizul Islam, G. P. Biswas, Muhammad Khurram Khan, and Neeraj Kumar. "A robust and anonymous patient monitoring system using wireless medical sensor networks." *Future Generation Computer Systems* 80 (2018): 483-495. https://doi.org/10.1016/j.future.2016.05.032

[2] Han, Fei, Jing Qin, Huawei Zhao, and Jiankun Hu. "A general transformation from KP-ABE to searchable encryption." *Future Generation Computer Systems* 30 (2014): 107-115. https://doi.org/10.1016/j.future.2013.09.013

[3] Das, Ashok Kumar, Sherali Zeadally, and Mohammad Wazid. "Lightweight authentication protocols for wearable devices." *Computers & Electrical Engineering* 63 (2017): 196-208. https://doi.org/10.1016/j.compeleceng.2017.03.008

[4] El-Hajj, Mohammed, Ahmad Fadlallah, Maroun Chamoun, and Ahmed Serhrouchni. "A survey of internet of things (IoT) authentication schemes." *Sensors* 19, no. 5 (2019): 1141. https://doi.org/10.3390/s19051141

[5] Ferrag, Mohamed Amine, Leandros A. Maglaras, Helge Janicke, Jianmin Jiang, and Lei Shu. "Authentication protocols for internet of things: a comprehensive survey." *Security and Communication Networks* 2017 (2017). https://doi.org/10.1155/2017/6562953

[6] Liao, Yi-Pin, and Shuenn-Shyang Wang. "A secure dynamic ID based remote user authentication scheme for multi-server environment." *Computer Standards & Interfaces* 31, no. 1 (2009): 24-29. https://doi.org/10.1016/j.csi.2007.10.007

[7] Jiang, Qi, Jianfeng Ma, Chao Yang, Xindi Ma, Jian Shen, and Shehzad Ashraf Chaudhry. "Efficient end-to-end authentication protocol for wearable health monitoring systems." *Computers & Electrical Engineering* 63 (2017): 182-195. https://doi.org/10.1016/j.compeleceng.2017.03.016

[8] Kumar, Lokesh, Debdulal Saha, Shakeb A. Khan, Kamalendu Sengupta, and Tarikul Islam. "A medium-range hygrometer using nano-porous thin film of gamma-Al2O3 with electronics phase detection." *IEEE sensors journal* 12, no. 5 (2012): 1625-1632. https://doi.org/10.1109/JSEN.2011.2172979

[9] Liao, Yi-Pin, and Shuenn-Shyang Wang. "A secure dynamic ID based remote user authentication scheme for multi-server environment." *Computer Standards & Interfaces* 31, no. 1 (2009): 24-29. https://doi.org/10.1016/j.csi.2007.10.007

[10] Martínez-Peláez, Rafael, Francisco Rico-Novella, Cristina Satizábal, and J. Pomykala. "Efficient and secure dynamic ID-based remote user authentication scheme with session key agreement for multi-server environment." *Int. J. Netw. Secur. Its Appl* 2 (2010): 106-116. https://doi.org/10.5121/ijnsa.2010.2409

[11] Perera, Charith, Chi Harold Liu, Srimal Jayawardena, and Min Chen. "A survey on internet of things from industrial market perspective." *IEEE Access* 2 (2014): 1660-1679. https://doi.org/10.1109/ACCESS.2015.2389854

[12] Shi, Wenbo, and Peng Gong. "A new user authentication protocol for wireless sensor networks using elliptic curves cryptography." *International Journal of Distributed Sensor Networks* 9, no. 4 (2013): 730831. https://doi.org/10.1155/2013/730831

[13] Wu, Fan, Xiong Li, Lili Xu, Arun Kumar Sangaiah, and Joel JPC Rodrigues. "Authentication protocol for distributed cloud computing: An explanation of the security situations for Internet-of-Things-enabled devices." *IEEE Consumer Electronics Magazine* 7, no. 6 (2018): 38-44. https://doi.org/10.1109/MCE.2018.2851744

[14] Yoon, Eun-Jun, Eun-Kyung Ryu, and Kee-Young Yoo. "Further improvement of an efficient password based remote user authentication scheme using smart cards." *IEEE Transactions on Consumer Electronics* 50, no. 2 (2004): 612-614. https://doi.org/10.1109/TCE.2004.1309437

[15] Zhou, Lu, Xiong Li, Kuo-Hui Yeh, Chunhua Su, and Wayne Chiu. "Lightweight IoT-based authentication scheme in cloud computing circumstance." *Future Generation Computer Systems* 91 (2019): 244-251. https://doi.org/10.1016/j.future.2018.08.038

[16] Mirsaraei, AmirHossein Ghafouri, Ali Barati, and Hamid Barati. "A secure three-factor authenticationscheme for IoT environments." *Journal of Parallel and Distributed Computing* 169 (2022): 87-105. https://doi.org/10.1016/j.jpdc.2022.06.011

[17] Alimoradi, Pourya, Ali Barati, and Hamid Barati. "A hierarchical key management and authentication method for wireless sensor networks." *International journal of communication systems* 35, no. 6 (2022): e5076. https://doi.org/10.1002/dac.5076

[18] Azhdari, Mohammad Sadegh, Ali Barati, and Hamid Barati. "A cluster-based routing method with authentication capability in Vehicular Ad hoc Networks (VANETs)." *Journal of Parallel and Distributed Computing* 169 (2022): 1-23. https://doi.org/10.1016/j.jpdc.2022.06.009

[19] Barati, Ali, Ali Movaghar, and Masoud Sabaei. "RDTP: Reliable data transport protocol in wireless sensor networks." *Telecommunication Systems* 62 (2016): 611-623. https://doi.org/10.1007/s11235-015-0098-2

[20] Barati, A., S. J. Dastgheib, A. Movaghar, and I. Attarzadeh. "An effective fuzzy based algorithm to detect faulty readings in long thin wireless sensor networks." *International Journal on Technical and Physical Problems of Engineering (IJTPE)* 3, no. 1 (2012): 2077-3528.

[21] Khazaei, Ehsan, Ali Barati, and Ali Movaghar. "Improvement of fault detection in wireless sensor networks." In *2009 ISECS International Colloquium on Computing, Communication, Control, and Management*, vol. 4, pp. 644-646. IEEE, 2009. https://doi.org/10.1109/CCCM.2009.5267508

[22] Javanmardi, Saeed, Ali Barati, Seyyed Jalaleddin Dastgheib, and Iman Attarzadeh. "A novel approach for faulty node detection with the aid of fuzzy theory and majority voting in wireless sensor networks." *International Journal of Advanced Smart Sensor Network Systems* 2, no. 4 (2012): 1-10. https://doi.org/10.5121/ijassn.2012.2401

[23] Azhdari, Mohammad Sadegh, Ali Barati, and Hamid Barati. "A cluster-based routing method with authentication capability in Vehicular Ad hoc Networks (VANETs)." *Journal of Parallel and Distributed Computing* 169 (2022): 1-23. https://doi.org/10.1016/j.jpdc.2022.06.009

[24] Barati, Ali, Ali Movaghar, Masoud Sabaei, and Samira Modiri. "Reliable wireless sensor networks by using redundant residue number system." In *2013 International Conference on Advanced Computer Science and Electronics Information (ICACSEI 2013)*, pp. 488-491. Atlantis Press, 2013. https://doi.org/10.2991/icacsei.2013.119

[25] Modiri, Samira, Ali Movaghar, and Ali Barati. "Study of Error Controllability for the New Modulus." *Journal of Advanced Computer Science and Technology* 1, no. 4 (2012): 176-186.

[26] Barati, A., M. Dehghan, A. Movaghar, and H. Barati. "Improving fault tolerance in ad-hoc networks by using residue number system." *Journal of Applied Sciences* 8, no. 18 (2008): 3273-3278. https://doi.org/10.3923/jas.2008.3273.3278

[27] Li, X., Niu, J., Khan, M. K., & Liao, J. (2013, July). Security Analysis and Enhancement of a Dynamic Identity Based Authentication Scheme Using Smart Cards. In *2013 International Symposium on Biometrics and Security Technologies* (pp. 149-154). IEEE.

[28] Li, L. H., Lin, L. C., & Hwang, M. S. (2001). A remote password authentication scheme for multiserver architecture using neural networks. *IEEE Transactions on Neural Networks*, *12*(6), 1498-1504.

[29] Chien, H. Y., Jan, J. K., & Tseng, Y. M. (2002). An efficient and practical solution to remote authentication: smart card. *Computers & Security*, *21*(4), 372-375.

[30] Chang, C. C., & Kuo, J. Y. (2005, March). An efficient multi-server password authenticated key agreement scheme using smart cards with access control. In *19th International Conference on Advanced Information Networking and Applications (AINA'05) Volume 1 (AINA papers)* (Vol. 2, pp. 257-260). IEEE.

[31] Amin, R., Kumar, N., Biswas, G. P., Iqbal, R., & Chang, V. (2018). A light weight authentication protocol for IoT-enabled devices in distributed Cloud Computing environment. *Future Generation Computer Systems*, *78*, 1005-1019.

[32] Wu, F., Xu, L., Kumari, S., Li, X., Shen, J., Choo, K. K. R., ... & Das, A. K. (2017). An efficient authentication and key agreement scheme for multi-gateway wireless sensor networks in IoT deployment. *Journal of Network and Computer Applications*, *89*, 72-85.

[33] Liao, Y. P., & Wang, S. S. (2009). A secure dynamic ID based remote user authentication scheme for multi-server environment. *Computer Standards & Interfaces*, *31*(1), 24-29.

[34] Zhou, L., Li, X., Yeh, K. H., Su, C., & Chiu, W. (2019). Lightweight IoT-based authentication scheme in cloud computing circumstance. *Future generation computer systems*, *91*, 244-251.

[35] Blanchet, B., Smyth, B., Cheval, V., & Sylvestre, M. (2018). ProVerif 2.00: automatic cryptographic protocol verifier, user manual and tutorial. *Version from*, 05-16.