Proceedings of the International Conference on Intelligent Computing and Communication for Smart World (I2C2SW-2018)
14-15 December 2018

Multi-factor Based User Authentication Scheme for Lightweight IoT Devices

J. Gowthami
Department of IT
Nandha College of Technology
Erode, India.
mail2gowthamij@gmail.com

N. Shanthi
Department of CSE
Kongu Engineering College
Perundurai, India.
shanthi.moorthi@gmail.com

Abstract— **Authentication of legitimate user plays a prominent role in security of the Internet of Things (IoT) applications like smart home, smart cities, wearable devices etc. The services offered by IoT devices must be attainable from anyplace and anytime, only by the legitimate users. The existing authentication schemes for IoT applications are not secure, vulnerable to many attacks and also any illegal user may access the data of smart devices. This paper presents an authentication scheme which uses only simple algebraic operations like hash and XOR, and this makes the proposed authentication scheme more suitable for the lightweight environment of IoT. It uses multiple factors like user identity, password, and biometrics for authenticating the legal users. Also, the proposed scheme establishes mutual authentication and key establishment between the users and smart devices involved in the communication. The protocol is modeled using the Security Protocol Description Language and the verification is done using one of the formal verification tools–Scyther. The result from the Scyther tool confirms the security of the protocol and clarifies that it is robust against various known attacks and are more affordable for practical applications.**

Keywords— Internet of Things, authentication, lightweight, multi-factor authentication, Security Protocol Description Language, Scyther.

I. Introduction

The Internet of Things (IoT) is a novel communication paradigm that visualizes the future, in which the objects of regular day to day existence are equipped with several components that would make them communicate with each other and with the users, transferring the data over the internet. IoT utilizes the internet as its necessary part. The idea of IoT was presented by the scientist Ashton, in the year 1999. IoT is an interconnection of things like sensor gadgets, tags, smart objects, etc., which are connected among themselves and with the Internet. IoT finds application in various aspects such as connected homes, versatile medicinal services, elderly help, industrial automation, intelligent energy management, smart grids, automotive, traffic administration and numerous others [1, 2].

The advent of smart technologies have made tremendous advantages to the public but at the same time it has opened gateways to hackers for stealing the valuable information from the data transmitted among the smart devices. To overcome this, it is necessary to ensure that the users accessing the services of the smart device must have the authorized credentials. Authentication is the core basics of secure communication between the users and smart devices [3].

Authentication can be made based on the following different factors [4-6],

- With what the user has (Ownership factors like smart cards, smartphones, tokens, etc.)
- With what the user knows (Knowledge factors like passwords, patterns) and
- With what the user is (Inherence factors like biometrics etc.)

There are some requirements that are listed as essentials for acquiring an efficient authentication scheme of IoT remote users [7, 8]. They include,

- Lightweight security solution: A lightweight authentication protocol is only suitable for the nature of IoT nodes. In general the smart devices in the IoT network are resource constrained. The processing power, battery backup, memory, speed etc are the parameters needs to be considered .To overcome this, a lightweight security solution is to be established.
- Session Key agreement: A shared session key is needed to be setuped between the users and smart nodes. This session key is necessary for secure communication.
- Mutual authentication: The parties involved in the communication must verify the authenticity of each other for secure authentication.
- Multiple factors for authentication: Single-factor authentication schemes suffer from several disadvantages and are much easier to hack. Hence the use of multi-factor authentication including biometrics increases the security.

A. Related work

An enormous number of works has been done in the IoT field incorporating various authentication schemes having both advantages and disadvantages. An overview of the related works is given in this section.

Lee et al. [9] suggested a user authentication scheme which uses fingerprint and smart cards for authenticating the user. This scheme utilises the concept of El Gamal public key cryptosystem. It doesn’t involve the maintenance of a password table by the system. This is because this scheme stores only two secret keys and eliminates the storage of any other keys or password. This scheme can defy an impersonation by adding extra keys.

Ruhul et al. [10] proposed an authentication protocol design for distributed cloud environment. All the information on the private cloud servers can be accessed safely by the users using the smart card. To establish the security measures of this model, AVISPA tool and BAN logic are used. Proverif simulator is used to measure the security strengths and informal cryptanalysis made on the protocol confirms that it is secure against all possible security threats. Gowthami et al. [11] proposed an authentication scheme for wearable device application. This mechanism supports secure access of application to remote users using Fuzzy extractor.

Xue et al. [12] proposed a new lightweight authentication scheme using temporal-credentials. The protocol is embedded with mutual authentication and key agreement schemes. It only needs hash, XOR computations and uses the recursive style to achieve mutual authentication among the user, gateway and the sensor nodes.

Fatimah et al. [13] proposed that the recent technology in fingerprint acquisition is to use different types of digital cameras and especially the use of mobile phone cameras. The available fingerprint datasets can be adopted using digital cameras are all taken with built-in degradations due to specific lighting, different backgrounds, low camera quality, image compression, and many more.

A solution for authentication problem based on biometrics was presented and adapted for the Wireless Sensor Networks (WSN) environment by Santoso et al. [14].It involves only simple lightweight calculations. The major advantage of this protocol is that the user's iris is used for authentication. To dramatically increase the security of the protocol user's key is generated every time from the user's iris.

The security of An's scheme [15] was examined by Das and Goswami [16] and the following shortcomings have been reported: (i) during the login phase, there is a flaw in user's biometric verification, (ii) during the login and authentication phases, there is a flaw in user's password verification, and (iii) user's password can be changed by the user at any time. In order to overcome the drawbacks in An's scheme, a new scheme is proposed which exploits smart card and biometrics for authenticating the remote user.

Generic framework of three-factor authentication to protect services and resources from unauthorized use has been proposed by Huang et al. [17]. The authentication is based on the password, smart card, and biometrics. The analysis shows that the framework satisfies all security requirements on three-factor authentication and has several other practice-friendly properties (e.g., key agreement, forward security, and mutual authentication).

Mittal et al. [29] presented the cryptanalysis and failings of Wang and Li's user authentication which utilized the fingerprint for authenticating the legal users. It is vulnerable to replay attack, Denial of Service (DoS) attack, and impersonation attack. Also they have suggested an enhanced fingerprint-based authentication scheme that could eliminate all the discovered drawbacks of Wang and Li's scheme.

Though there are many works on the authentication of lightweight devices of IoT, they are not assured to be robust against various known attacks. Hence a robust multi-factor based authentication scheme establishing mutual authentication and session key agreement has been proposed in this paper.

II. Preliminaries

This section gives a brief overview of one-way hash function and perceptual hash, which are used in the proposed scheme.

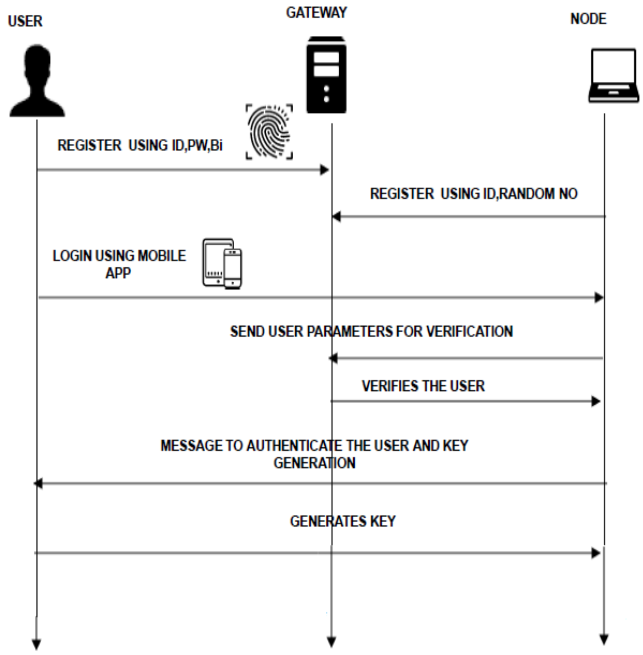

USER
GATEWAY
NODE
REGISTER USING ID,PW,Bi
REGISTER USING ID,RANDOM NO
LOGIN USING MOBILE APP
SEND USER PARAMETERS FOR VERIFICATION
VERIFIES THE USER
MESSAGE TO AUTHENTICATE THE USER AND KEY GENERATION
GENERATES KEY

Fig. 1 Authentication model

A. Network model

The network model for the proposed scheme is given in Fig.1. This proposed network model consists of a single gateway node and different smart nodes residing in the IoT network and many numbers of users. The gateway node acts as a bridge between the users and the smart nodes. The user U_i and smart nodes N_j must first register themselves with the gateway node. U_i uses identity IDi, a random nonce, password PWi and biometrics Bi for registration. N_j uses a random nonce, identity NIDj for their registration with the gateway node. After the successful registration, the users can get the services of the smart nodes by logging in, with the help of their passwords, biometrics and identity. The smart nodes will authenticate the user only after the user validation by the gateway node. After the successful authentication of the user, smart nodes generate a session key for secured communication.

B. One-way hash function

A one-way hash function is a mathematical function which takes a variable-length string as input and produces a fixed-length binary sequence as output. Moreover, the one-way hash function is very difficult to reverse i.e., it is very difficult to find a string which could hash to the given value. It is also called as message digest, fingerprint or compression function. A good hash function will make hard to find two input strings that would hash to the same value. The recent hash functions produce hash values of 128 bits or higher and used to create digital signatures [18].

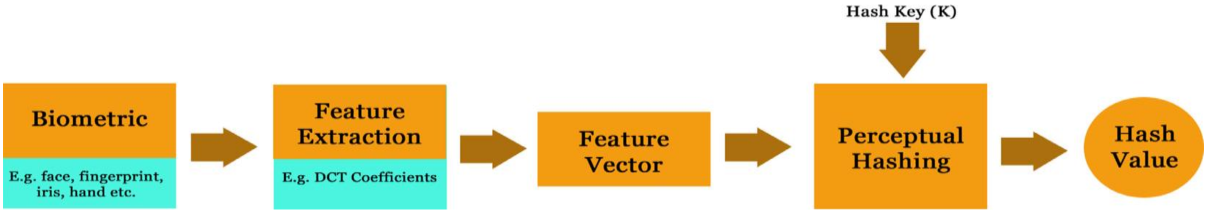


Fig. 2 Perceptual hash process

C. Perceptual hash

The standard encryption techniques or normal hash algorithms can't be used for extracting keys from biometrics. The reason for this is that the biometric data such as fingerprint, voice, etc. varies with time and other external factors. To overcome this problem, perceptual hash (P-Hash) can be used. The perceptual hash function produces same hash value even if the content gets any significant modification. They are designed especially to address the difference in quality and formats, i.e., the same biometric input should always maps to the same hash value. The measure of the hash value generated by perceptual hash ranges from 64 bits to 128 bits [ref 8, 28]. The perceptual hash function is shown in Fig. 2

III. Proposed Scheme

The lightweight authentication scheme can be proposed with the following phases-User registration phase, Smart node registration phase, Login phase, Authentication phase, Biometrics and password change phase. The notations used for explaining the proposed scheme are given in Table 1.

TABLE I. Notations

Notations	Description
User	*Ui*
Gateway	*GWN*
Node	*Nj*
⊕	*XOR* operation
H (.)	One way hash function
h(.)	Perceptual hash function
Ni, Nj	Nonce
Ti	Current timestamp
ΔT	Maximum transmission delay
||	Concatenation operation
IDi, PWi, Bi	*Ui*'s Identity, password, personal biometrics
Sgu	Secret parameter available for *GW* and *Ui*
Sgn	Secret parameter available for *GW* and *Nj*
Sg	Secret parameter given only to *GW*
SK	Shared key for each session
HIi	Hashed Identity
UNi, Uzi, Aj, Fij	User calculated parameter
HPi	Hashed password
HBi	Hashed Biometrics

A. User Registration Phase

The registration of User Ui with the Gateway Node GWN takes place in this phase. Fig.3 shows the steps involved in this phase.

U1: Ui generates a random nonce ni and calculates the hash value of identity IDi, password PWi, biometrics Bi using the operations HPi=H(ni||PWi), HIi=H(ni||IDi), HBi=h(ni||IBi) respectively. The Ui will then send the request parameters <HPi, HIi, HBi> to the gateway node for registration.

U2: GWN, on receiving the registration request from the Ui will perform some operations on the received parameters like Xi=H(HIi||Sg), Yi=H(HPi||Sgu), Zi=H(HBi||Sgu), Ei=Xi⊕Yi, Fi=Xi⊕Zi and send HBi, Ei, Fi, Xi, Sgu to the user and these parameters are stored in Ui's memory.

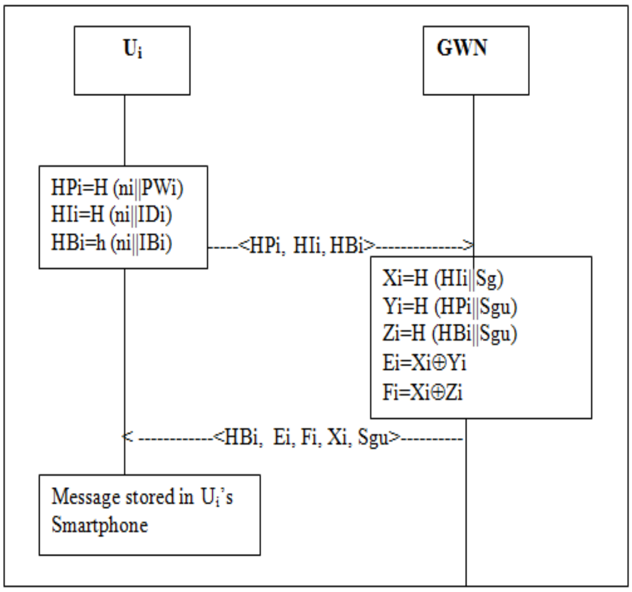

U_i
GWN
HPi=H (ni||PWi)
HIi=H (ni||IDi)
HBi=h (ni||IBi)
-----<HPi, HIi, HBi>--------------->
Xi=H (HIi||Sg)
Yi=H (HPi||Sgu)
Zi=H (HBi||Sgu)
Ei=Xi⊕Yi
Fi=Xi⊕Zi
< ------------<HBi, Ei, Fi, Xi, Sgu>----------
Message stored in U_i's Smartphone

Fig.3 User Registration Phase

B. Smart Node Registration Phase

There may be several smart devices connected to the IoT network. Below are the steps for registering the Smart node with the Gateway node. Fig.3 shows the Smart node registration Phase.

SR1: The node N_j chooses a random nonce nj, Identity NIDj and has a secret key Sgn, which is known only to the GWN and N_j.

SR2: The Node N_j performs XOR operations MPj=H (Sgn||nj||NIDj), MNj= nj⊕sgn, Pj= Mpj ⊕ MNj and sent the parameters <NIDj, RMPj, MNj, TS1> to GWN.

SR3: GWN on receiving the request from the Smart node, assures whether the received timestamp TS1 from the node is less than or equal to the transmission delay ΔT. If the above condition is not met then the registration request is declined.

SR4: On satisfaction of the condition, GWN will then perform the algebraic operations $MPj=RMPj\oplus MNj$, $Nj^*=Sgn\oplus MNj$, $Mpj^*=H(Sgn\| nj^*\|NIDj)$. GWN will validate the equality of MPj*&MPj. If so $Xj=H(NIDj\ \|Sg)$, $Yj=H(MPj\|Sgn)$, $Ej=Xj\oplus Yj$ are calculated. GWN will send the response message <Ej, Xj, TS2 > to the N_j.

SR5: N_j again checks for timestamp $|TS2-T|<\Delta T$. If the condition for the timestamp is not satisfied then the transmission delay time interval ΔT is greater than the actual timestamp. In such a case N_j will reject the parameters. If the condition is satisfied then Ej, Xj is stored in memory.

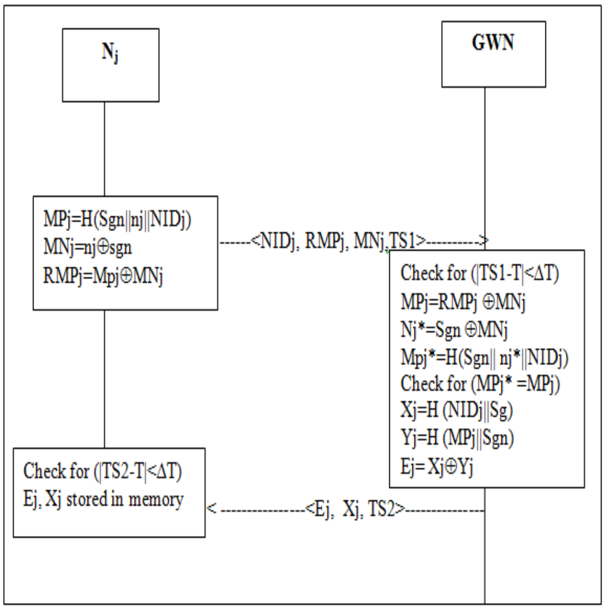


Fig.4 Smart node Registration Phase

C. Login Phase

After the successful completion of the registration phase, Ui can connect to any desired device in the IoT environment through the login phase. The User must enter his/her credentials like Password, biometrics and Identity to access the services of any smart device. The following is to be followed, for login to get the services of the desired device. Fig.5 shows the login phase of the proposed protocol

L1: The user, Ui first enter the Password PWi* and Biometrics Bi* for verification. The following operations are carried out for the verification process. $HPi= H(ri\|PWi^*)$, $HBi=H(ri\|Bi^*)$,$Yi=Xi\oplus Ei$, $Zi=Xi\oplus Fi$.

L2: Ui will then verify whether $Yi^*= Yi$ && $Zi^*=Zi$ are equal. If yes, then $UNi= H(yi\|Zi\ \|Sgu\|TS1)$, $Uzi=n\oplus Xi$. If No, the login process is stopped.

L3: Finally, the parameters<BIi, Ei, Fi, Uzi, UNi, TS1> are sent to Nj, for the user to get connection with the node N_j.

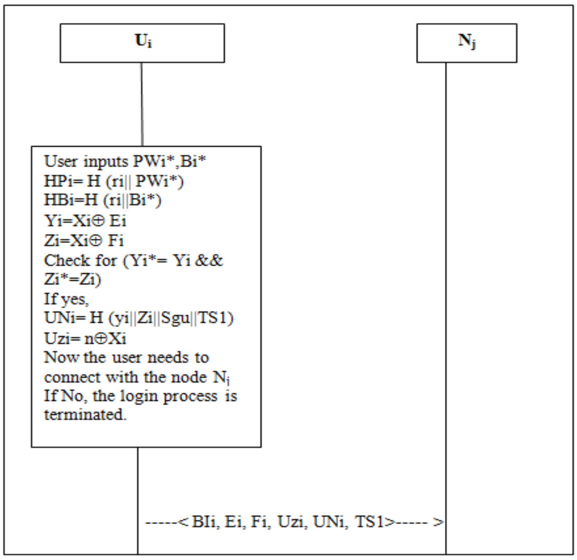


Fig.5 Login Phase

D. Authentication Phase

In this Phase, the parameters of the users are verified by the Gateway node. Also, N_j is encouraged to start the communication with the user. Fig.6 gives the detailed steps involved in the authentication phase.

A1: In the authentication phase, N_j on receiving the parameters < BIi, Ei, Fi, Uzi, UNi, TS1> will check for the correctness of timestamp. Then the value of Yj and Aj are calculated from $Yj= Xj\oplus Ej$, $Aj=H(Sgn\|TS1\|TS2)\oplus Yj$. The resultant parameters <BIi, Ei, Fi, UNi, NIDj, Ej, Aj, TS2, TS1> are sent to the GWN.

A2: GWN would check for timestamp condition $|TS2-T|<\Delta T$. If this is satisfied then it can be considered that there will be less transmission delay ΔT. If it so, GWN will compute $Xj^*= H(NIDj\|Sg)$, $Yj^*=Ej\oplus Xj^*$, $Yj=Aj\oplus H(Sgn\|TS1\|TS2)$.

A3: GWN will then check for the condition $Yj^*==Yj$. If the condition holds, then the values $Xi^*=H(HIi\|Sg)$, $Yi^*=Ei\oplus Xi^*$, $Qi=H(Yi^*\|Sgu\|TS1\|Zi^*)$ are calculated. GWN will then check for the condition $UNi==Qi$. If the condition again holds, then $Pij=Xi^*\oplus H(Xj\|Sgn)$, $Hj=H(Xj^*\|Sgn\|TS1\|TS2\|TS3)$, $Vi=H(Qi\|TS1\|TS2\|\ TS3)\ \|\ b_i^*\|c_i^*\|T_1)$ are calculated. GWN then sent < Pij, Hj, Vi, TS1, TS2, TS3> to the N_j.

A4: N_j will check $(Hj==H(Xj\|Xgn\|TS1\|TS2\|TS3))$ $Xj^*=Pij^*\oplus H(Xj\|Xgn)$. If the condition holds well, then Session key is generated with nonce m and $n=UZi\oplus Xj^*$. Finally value of session key, $SK=H(n\oplus m)$ and parameters <Rij, Nj, TS1, TS2, TS3, TS4, Vi> are sent to Ui.

A5: Ui will verify for the validation of the received parameters by verifying the timestamp and then $[Vi== H(H((Ei\oplus Xi)\|TS1)\|TS2\|TS3)]$. After the verification, Ui calculates $n=Rij\oplus H(Xi\|NIDj\|$

TS1||TS2||TS3||TS4), SK= (n⊕m). With this session key, Communication can take place between the users and the smart devices in the IoT network.

E. Biometrics and Password Change Phase

This phase allows the users to easily update their password and biometrics. It is necessary for a user to give his/her old password and biometrics to set the new password and new biometrics. A regular updating of the password and biometrics is necessary to increase the security.Fig.7 gives the diagrammatic representation of steps involved in this phase.

P1: The user Ui first gives the existing password PWi* and biometrics Bi*.

P2: The user calculates the hashed form of password and biometrics with HPi*=H (ni||PWi*), HBi*=H (ni*||Bi*). Then Ui calculates Yi*=H (HPi*||Sgu), Zi*=H (HBi*||Sgu).

P3: Ui will then get the stored values Xi, Ei, Fi and calculates the values of Yi, Zi using Yi=Xi⊕Ei, Zi= Zi⊕Fi. After then, Ui will check for the equality of (Yi==Yi*) && (Zi==Zi*).

P4: Ui can now give the updated Password NPWi and Biometrics NBi. The Ui calculates updated value of Ei as NEi with NHPi= H (ni||NPWi), NYi=H (NHPi||Sgu), NEi= Xi⊕NYi and new value of Fi as NFi with NHBi=H (ni||NBi), NZi=H (NHBi||Sgu) and NFi=Xi⊕NZi.

P5: Finally, old Ei can be replaced with NEi and old Fi can be replaced with NFi.

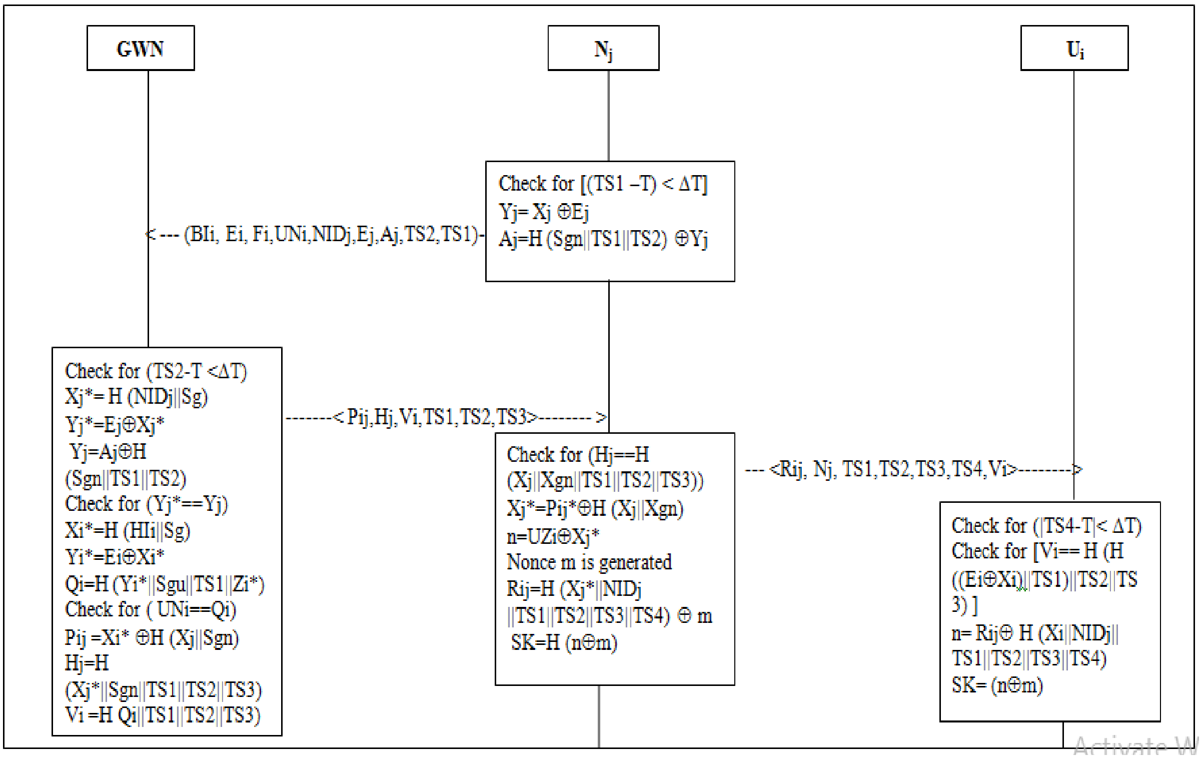
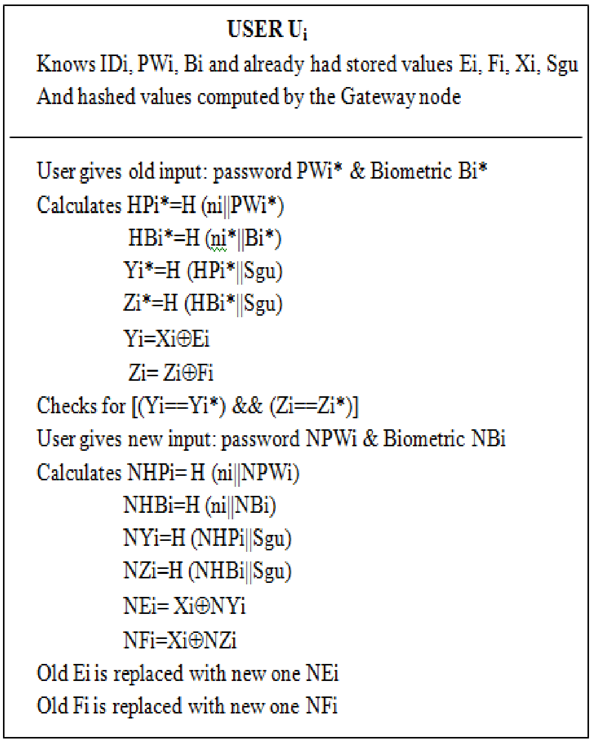


Fig.6 Authentication Phase

IV. VERIFICATION USING SCYTHER

Scyther and Scyther GUI were evolved by CAS Cremer in 2007. Scyther GUI integrates the Scyther command-line tool and python scripting interfaces. Scyther tool's inputs are the protocol to be validated and parameters which is optional and outputs the report summary with the graph generated for each attack.

Scyther is a tool used for confirmation of security of network protocols. The resultant protocol from the Scyther tool is assumed to have perfect cryptographic functions. The Graphical User Interface (GUI) provided by this tool makes it simple to collaborate and have knowledge of the protocols. Moreover, graphs would be generated for each of the attack found in the given protocol, corresponding to the mentioned claim. The Scyther tool also traces the possible claims of the protocol. It also has the capability to detect during the construction of any desired protocol. The trace patterns could be generated under certain criteria. Either under bounded or an unbounded number of sessions Scyther would execute the protocol validation [19].



USER Ui

Knows IDi, PWi, Bi and already had stored values Ei, Fi, Xi, Sgu
And hashed values computed by the Gateway node

User gives old input: password PWi* & Biometric Bi*
Calculates HPi*=H (ni||PWi*)
HBi*=H (ni*||Bi*)
Yi*=H (HPi*||Sgu)
Zi*=H (HBi*||Sgu)
Yi=Xi⊕Ei
Zi= Zi⊕Fi
Checks for [(Yi==Yi*) && (Zi==Zi*)]
User gives new input: password NPWi & Biometric NBi
Calculates NHPi= H (ni||NPWi)
NHBi=H (ni||NBi)
NYi=H (NHPi||Sgu)
NZi=H (NHBi||Sgu)
NEi= Xi⊕NYi
NFi=Xi⊕NZi
Old Ei is replaced with new one NEi
Old Fi is replaced with new one NFi

Fig.7 Biometrics and password change phase

Security Protocol Description Language (SPDL)

To verify a protocol in Scyther tool, it must be written in a language called Security Protocol Description Language (SPDL). Security properties are modeled as claim events. The syntax of Scyther's language SPDL is slightly like C and Java. The SPDL would mainly define the core properties of the protocols by using set of roles. Roles are nothing but the sequence of events and most events describe the process of the sending or receiving of terms.Fig.8 shows the prototype of the Scyther tool.

The following three ways are used by the Scyther tool, for the protocol validation [20],

1) Claim verification:
Scyther would verify the given protocol and results whether the protocol satisfies or falsifies the security properties.
2) Claim Automation:
If the security properties of the protocol are not specified by the user as claim events, then the claim events would be automatically generated and verified by the Scyther.
3) Characterization :
Each and every role in the protocol can be "characterized".

Scyther validates the provided protocol and provides a finite representation of all traces that contain an execution of the protocol role. Scyther would generate attack graph for the traced security vulnerabilities. For each of the claim mentioned, Scyther represents individual attack graph.

A. *Specification Of Protocol*

The proposed scheme is modeled in Scyther by using 3 roles - User U, the gateway Node G and the Node N. Scyther will stop the verification process if any attack is found and attack graph will be generated for the corresponding attacks found. The security properties of the proposed scheme can be given in form of claims in SPDL.

There are several initial declarations, including fresh nonce and other variable nonce received. The communication between the parties takes place with the help of send() and recv() commands. Security properties to be verified are given in the form of claims.

Following are the claims for User U

- claim_u1 (U, Secret, k (U, G)) - K(U, G) is the key known to the user and is kept unknown to the adversary.
- claim_g2 (G, Alive)- Alive is a form of authentication which aims to ensure that G has executed some events
- claim_g3 (G, Weak agree) -to check for non-injective agreement.
- claim_u4 (U, Niagree) – to check for non-injective synchronization

The main aim of SPDL language is to design protocols with some set of roles. Roles are defined by a sequence of events. These events mostly represent the sending and receiving actions of messages. Fig.9, 10 and 11 give the design of the proposed protocol in SPDL language. Fig. 9 represents User part of Protocol; Fig.10 represents Smart nodes part of Protocol in SPDL. Fig. 11 gives the Gateway node's part in the Protocol;

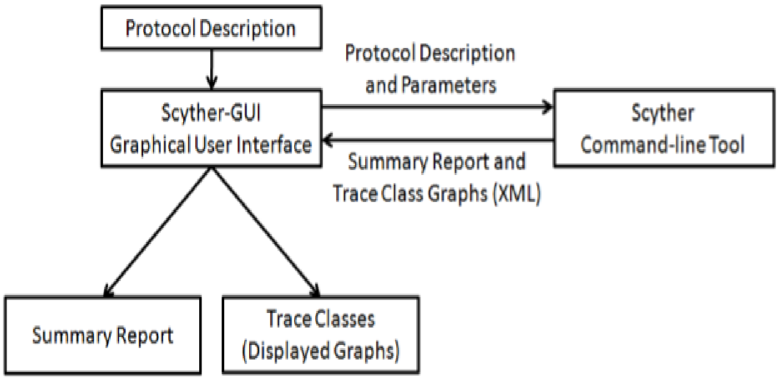
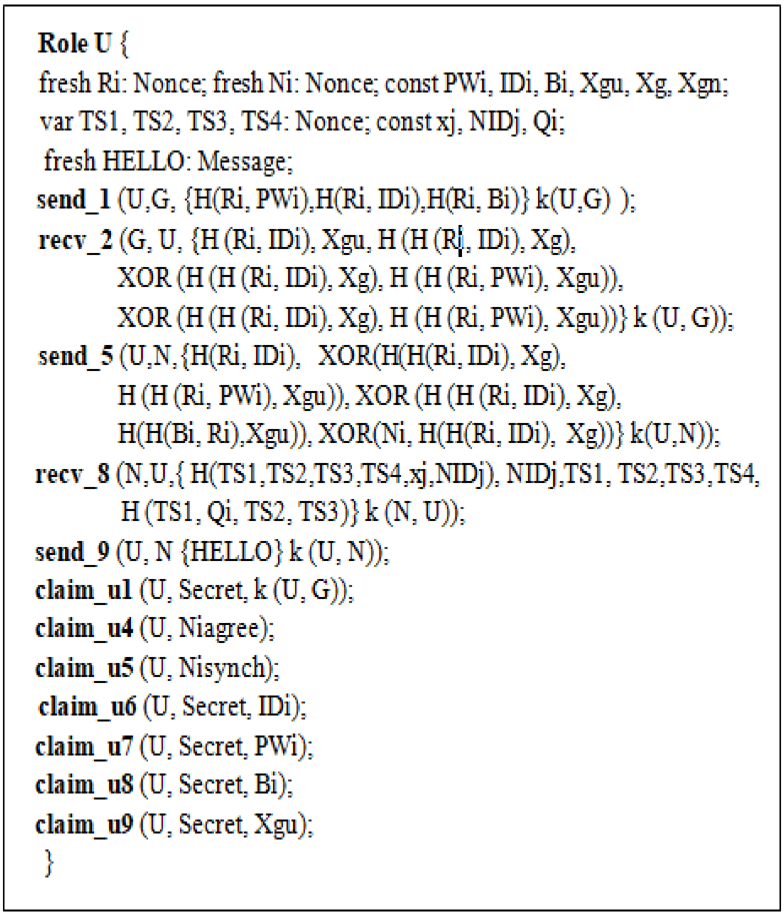
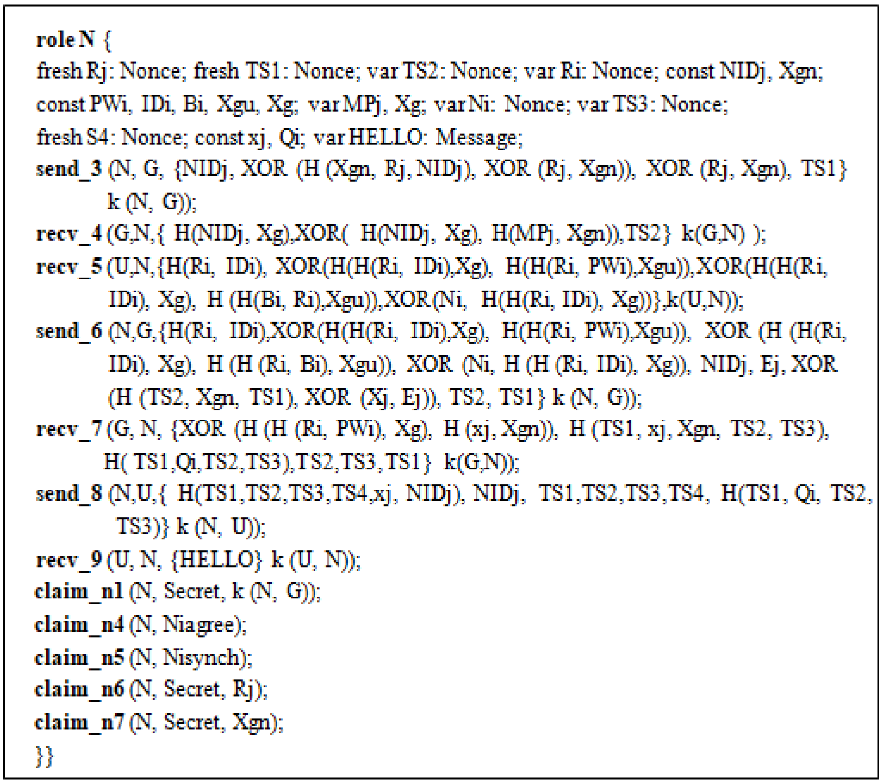

Protocol Description
Protocol Description and Parameters
Scyther-GUI Graphical User Interface
Scyther Command-line Tool
Summary Report and Trace Class Graphs (XML)
Summary Report
Trace Classes (Displayed Graphs)

Fig.8 Prototype of Scyther



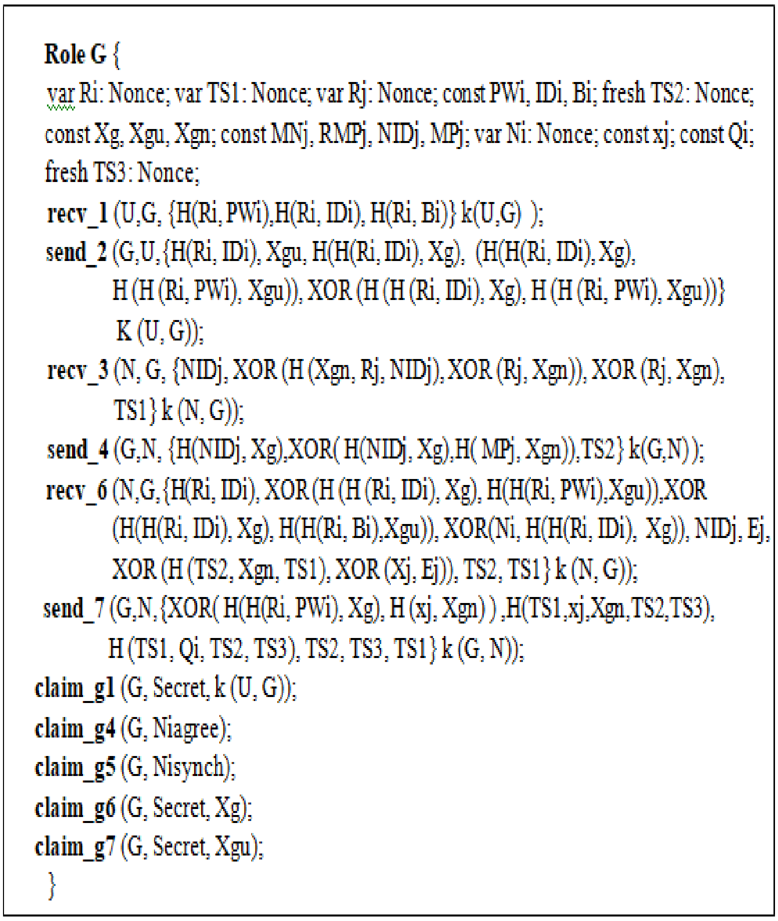
```
Role U {
fresh Ri: Nonce; fresh Ni: Nonce; const PWi, IDi, Bi, Xgu, Xg, Xgn;
var TS1, TS2, TS3, TS4: Nonce; const xj, NIDj, Qi;
 fresh HELLO: Message;
send_1 (U,G, {H(Ri, PWi),H(Ri, IDi),H(Ri, Bi)} k(U,G) );
recv_2 (G, U, {H (Ri, IDi), Xgu, H (H (Ri, IDi), Xg),
         XOR (H (H (Ri, IDi), Xg), H (H (Ri, PWi), Xgu)),
         XOR (H (H (Ri, IDi), Xg), H (H (Ri, PWi), Xgu))} k (U, G));
send_5 (U,N,{H(Ri, IDi),  XOR(H(H(Ri, IDi), Xg),
         H (H (Ri, PWi), Xgu)), XOR (H (H (Ri, IDi), Xg),
         H(H(Bi, Ri),Xgu)), XOR(Ni, H(H(Ri, IDi),  Xg))} k(U,N));
recv_8 (N,U,{ H(TS1,TS2,TS3,TS4,xj,NIDj), NIDj,TS1, TS2,TS3,TS4,
          H (TS1, Qi, TS2, TS3)} k (N, U));
send_9 (U, N {HELLO} k (U, N));
claim_ul (U, Secret, k (U, G));
claim_u4 (U, Niagree);
claim_u5 (U, Nisynch);
 claim_u6 (U, Secret, IDi);
claim_u7 (U, Secret, PWi);
claim_u8 (U, Secret, Bi);
claim_u9 (U, Secret, Xgu);
 }
```

Fig. 9 Protocol specification for User



```
role N {
fresh Rj: Nonce; fresh TS1: Nonce; var TS2: Nonce; var Ri: Nonce; const NIDj, Xgn;
const PWi, IDi, Bi, Xgu, Xg; var MPj, Xg; var Ni: Nonce; var TS3: Nonce;
fresh S4: Nonce; const xj, Qi; var HELLO: Message;
send_3 (N, G, {NIDj, XOR (H (Xgn, Rj, NIDj), XOR (Rj, Xgn)), XOR (Rj, Xgn), TS1}
         k (N, G));
recv_4 (G,N,{ H(NIDj, Xg),XOR(  H(NIDj, Xg), H(MPj, Xgn)),TS2}  k(G,N) );
recv_5 (U,N,{H(Ri,  IDi), XOR(H(H(Ri,  IDi),Xg),  H(H(Ri, PWi),Xgu)),XOR(H(H(Ri,
         IDi), Xg),  H (H(Bi, Ri),Xgu)),XOR(Ni,  H(H(Ri, IDi), Xg))},k(U,N));
send_6 (N,G,{H(Ri,  IDi),XOR(H(H(Ri,  IDi),Xg),  H(H(Ri, PWi),Xgu)),  XOR (H (H(Ri,
         IDi), Xg),  H (H (Ri, Bi), Xgu)),  XOR (Ni, H (H (Ri, IDi), Xg)),  NIDj, Ej, XOR
         (H (TS2, Xgn, TS1), XOR (Xj, Ej)), TS2, TS1} k (N, G));
recv_7 (G, N, {XOR (H (H (Ri, PWi), Xg),  H (xj, Xgn)),  H (TS1, xj, Xgn,  TS2, TS3),
        H( TS1,Qi,TS2,TS3),TS2,TS3,TS1}  k(G,N));
send_8 (N,U,{  H(TS1,TS2,TS3,TS4,xj, NIDj), NIDj,  TS1,TS2,TS3,TS4,  H(TS1, Qi, TS2,
          TS3)} k (N, U));
recv_9 (U, N, {HELLO} k (U, N));
claim_nl (N, Secret, k (N, G));
claim_n4 (N, Niagree);
claim_n5 (N, Nisynch);
claim_n6 (N, Secret, Rj);
claim_n7 (N, Secret, Xgn);
}}
```

Fig. 10 Protocol specification for Smart Node



```
Role G {
var Ri: Nonce; var TS1: Nonce; var Rj: Nonce; const PWi, IDi, Bi; fresh TS2: Nonce;
const Xg, Xgu, Xgn; const MNj, RMPj, NIDj, MPj; var Ni: Nonce; const xj; const Qi;
fresh TS3: Nonce;
recv_1 (U,G, {H(Ri, PWi),H(Ri, IDi), H(Ri, Bi)} k(U,G) );
send_2 (G,U,{H(Ri, IDi), Xgu, H(H(Ri, IDi), Xg), (H(H(Ri, IDi), Xg),
        H (H (Ri, PWi), Xgu)), XOR (H (H (Ri, IDi), Xg), H (H (Ri, PWi), Xgu))}
         K (U, G));
recv_3 (N, G, {NIDj, XOR (H (Xgn, Rj, NIDj), XOR (Rj, Xgn)), XOR (Rj, Xgn),
        TS1} k (N, G));
send_4 (G,N, {H(NIDj, Xg),XOR( H(NIDj, Xg),H( MPj, Xgn)),TS2} k(G,N) );
recv_6 (N,G,{H(Ri, IDi), XOR (H (H (Ri, IDi), Xg), H(H(Ri, PWi),Xgu)),XOR
        (H(H(Ri, IDi), Xg), H(H(Ri, Bi),Xgu)), XOR(Ni, H(H(Ri, IDi), Xg)), NIDj, Ej,
        XOR (H (TS2, Xgn, TS1), XOR (Xj, Ej)), TS2, TS1} k (N, G));
send_7 (G,N,{XOR( H(H(Ri, PWi), Xg), H (xj, Xgn) ) ,H(TS1,xj,Xgn,TS2,TS3),
        H (TS1, Qi, TS2, TS3), TS2, TS3, TS1} k (G, N));
claim_g1 (G, Secret, k (U, G));
claim_g4 (G, Niagree);
claim_g5 (G, Nisynch);
claim_g6 (G, Secret, Xg);
claim_g7 (G, Secret, Xgu);
 }
```

Fig. 11 Protocol specification for Gateway

B. *Result Analysis*

Fig.12 shows the simulation result for each role (U, G, N) involved in the proposed protocol using Scyther tool. It can be seen from Fig. 12 that Scyther's automatic security claims have been successfully verified. Also, it is to be noted that no attacks are found on the proposed authentication scheme and hence it is displayed as no attacks within bounds

V. PERFORMANCE COMPARISON

This section gives the comparison of the Communication cost (in bits), Computational time (in ms) and the security properties of the proposed scheme with the other schemes.

A. Computational time analysis

For calculating the computation time, the existing experimental values can be used. The notations T_{exp}, T_E/ T_D, T_h $T_{fe,}$ T_{mac} T_{hmac} are used to refer the computation time of modular exponentiation operation, symmetric encryption/decryption, hash function h(·), Gen(·)/Rep(·), message authentication code (MAC), hashed MAC respectively. The approximate value for T_{exp}, T_E/T_D, $T_{h,}$ T_{fe}, T_{mac} ,T_{hmac} assumed to be 0.0192, 0.0056, 0.00032, 0.0171, 0.0056 and 0.0056 respectively. Table 2 gives the comparison of the proposed with the other existing schemes.

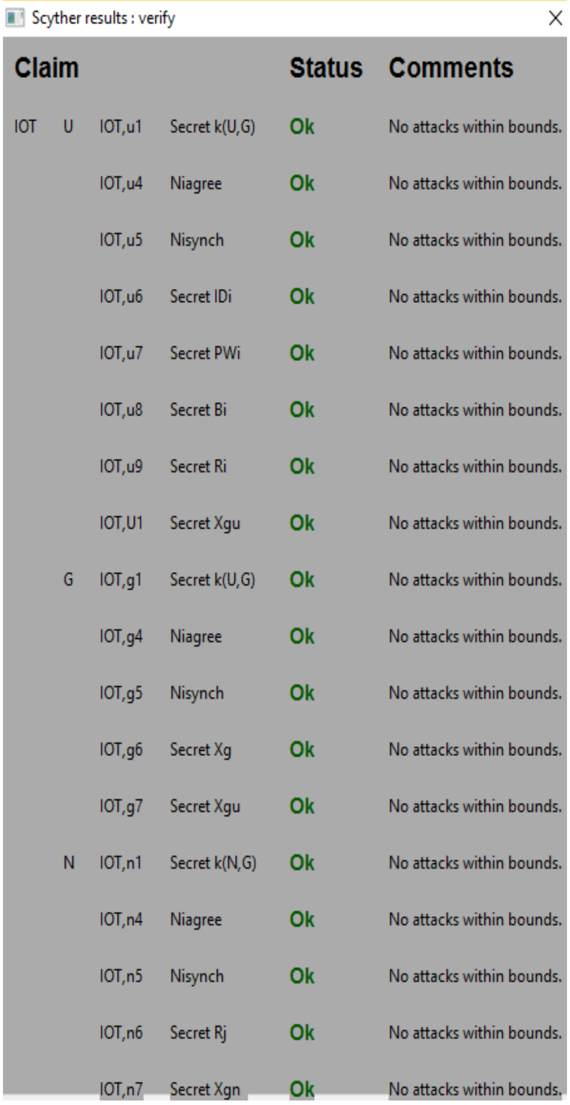
Fig.13 gives the total time (in ms) required for several authentication schemes. It is clear from the Fig.13that the time required for computation is very low for the proposed scheme, in comparison with others.

B. Communication cost analysis

For the calculation of Communication cost some reasonable assumptions are made. The communication cost analysis of the proposed and the other existing schemes which are considered for comparison are given in Table 3. To do so, the following are considered.

- Sequence number, random nonce or time stamp is of length 32 bits.
- Hash digest length is 160 bits- Hash function used is secure hash standard (SHA-1)[27]
- Identity ID is of length 160 bits.

Fig.14 gives the total cost (in bits) required for several authentication schemes. It is clear from the Fig.14 the cost required for the communication is low for the proposed scheme in comparison with others



Claim				Status	Comments
IOT	U	IOT,u1	Secret k(U,G)	Ok	No attacks within bounds.
		IOT,u4	Niagree	Ok	No attacks within bounds.
		IOT,u5	Nisynch	Ok	No attacks within bounds.
		IOT,u6	Secret IDi	Ok	No attacks within bounds.
		IOT,u7	Secret PWi	Ok	No attacks within bounds.
		IOT,u8	Secret Bi	Ok	No attacks within bounds.
		IOT,u9	Secret Ri	Ok	No attacks within bounds.
		IOT,U1	Secret Xgu	Ok	No attacks within bounds.
	G	IOT,g1	Secret k(U,G)	Ok	No attacks within bounds.
		IOT,g4	Niagree	Ok	No attacks within bounds.
		IOT,g5	Nisynch	Ok	No attacks within bounds.
		IOT,g6	Secret Xg	Ok	No attacks within bounds.
		IOT,g7	Secret Xgu	Ok	No attacks within bounds.
	N	IOT,n1	Secret k(N,G)	Ok	No attacks within bounds.
		IOT,n4	Niagree	Ok	No attacks within bounds.
		IOT,n5	Nisynch	Ok	No attacks within bounds.
		IOT,n6	Secret Rj	Ok	No attacks within bounds.
		IOT,n7	Secret Xgn	Ok	No attacks within bounds.

Fig. 12 Scyther results

TABLE II. COMPUTATIONAL TIME ANALYSIS

Scheme	Total cost	Estimated time(ms)
Kim-Kim [21]	$30T_h + 3T_E/T_D$	26.40
Sood et al.[22]	$24T_h$	7.68
Yoon- Yoo [23]	$17t_h + 4t_{epm}$	73.84
Kumari et al.[24]	$8_{th} + 4t_{ch} + 1_{tf}$	103.74
Vaidya et al. [25]	$20T_h + 3T_E/T_D$	23.20
Proposed scheme	$22\ T_h$	7.04

TABLE III. COMMUNICATION COST COMPARISON

Scheme	Total Messages	Total cost (in bits)
Kim-Kim [21]	2	4352
Sood et al.[22]	4	2400
Yoon- Yoo [23]	5	2880
Kumari et al.[24]	3	3296
Vaidya et al. [25]	2	2272
Proposed scheme	4	2144

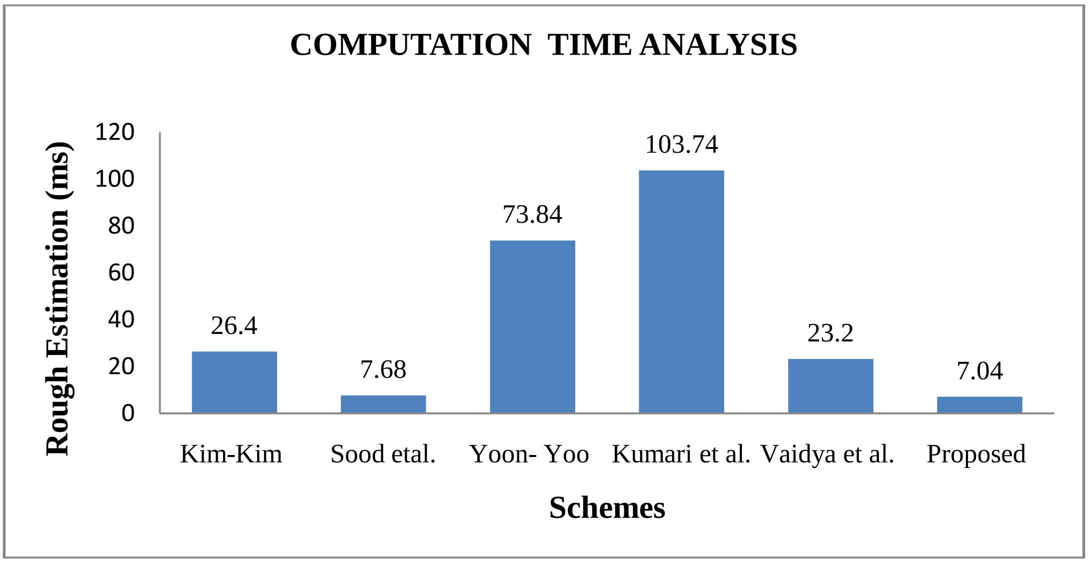


Fig. 13 Comparison of Computation time

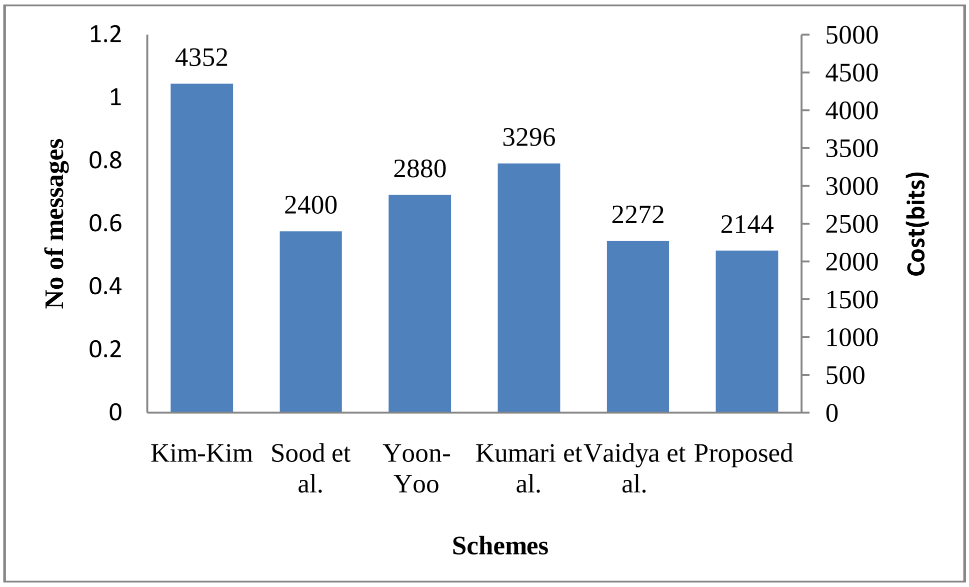


Fig. 14 Comparison of Communication cost

TABLE IV. Security Feature Comparison

Functional features	**[21]**	**[22]**	**[23]**	**[24]**	**[25]**	**Proposed scheme**
Session Key agreement	✓	✓	✓	✓	✓	✓
Parallels session attack	×	×	×	×	×	✓
Replay attack	✓	✓	✓	✓	✓	✓
User anonymity	×	✓	×	✓	×	✓
Mutual authentication	✓	×	×	✓	✓	✓
Impersonation attack	✓	✓	✓	✓	✓	✓
Man-in-the-middle attack	✓	×	×	✓	✓	✓
Offline guessing attack	×	✓	✓	✓	✓	✓
Gateway node bypassing attack	✓	×	×	×	✓	✓
Resilience to node capture attack	✓	×	✓	×	×	✓
Denial-of-service attack	✓	×	×	×	✓	✓
Biometric update phase	×	×	×	×	×	✓
Password update phase	×	×	✓	✓	✓	✓
Stolen smart device		×	✓	✓		✓

C. Comparison of security features

Table 4 gives the security features of the proposed scheme in comparison with other schemes. From table 4, it is clear that the proposed scheme is secure against several threats like Resilience to node capture attack, User anonymity, Parallel session attack, Impersonation attack etc.

VI. Conclusion

An authentication model utilizing multiple factors for authorizing the legal users have been presented in this paper. This protocol is best suitable for IoT networks since it uses only lightweight operations like XOR and hash. Also, the keys can be extracted inspite of the changes in the Biometrics with the help of Fuzzy extractor technique. The implementation of the protocol in the Scyther tool proves that it is secure against various known attacks like Impersonation attack, Eavesdropping attack, Man-in-the-middle attack, Password change attack etc. Moreover, the protocol agrees on mutual authentication and key agreement scheme. Further, in the future, the protocol can be tested in a hardware environment and any more optimization on the protocol can also be made.

References

[1] Zanella, A., Bui, N., Castellani, A. P., Vangelista, L., &Zorzi, M. Internet of things for smart cities, IEEE Internet of Things Journal, 2014, 1(1), pp. 22-32.

[2] Zaslavsky, A.B., Perera, C. &Georgakopoulos, D. Sensing as a service and big data, In Proceedings of International Conference on Advances in Cloud Computing: ACC-2012, 2012, pp.219.

[3] Gomez, C. &Paradells (2010).Wireless home automation networks: A survey of architectures and technologies. IEEE Communications Magazine, 48(6), 92–101.

[4] Gigli, M. & Koo S. Internet of things: services and applications categorization. Advances in Internet Things, 2011; 1(02):27–41. doi: 10.1016/j.future.2015.09.016 .

[5] Choi, Y., Nam, J., Lee, D., Kim, J., Jung, J. & Won, D. Security-enhanced anonymous multi server authenticated key agreement scheme using smart cards and biometrics. Sci World J., 2014, doi: 10.1155/2014/281305, https://www.hindawi.com/ journals/tswj /2014/281305/

[6] Liu, M. & Shieh, WG. On the security of Yoon and Yoo's biometrics remote user authentication scheme. WSEAS Trans Inf Sci Appl, 2014, 11, 94– 103.

[7] Yang, D & Yang, B. A biometric password-based multi-server authentication scheme with a smart card. In Proceedings of International Conference on Computer design and applications (ICCDA), IEEE, 2010, 5,540–54. doi: 10.1109/ICCDA.2010. 5541128

[8] Dhillon, P.K. & Sheetal Kalra. A lightweight biometrics based remote user authentication scheme for IoT services, Journal of Information Security and Applications, 2017, 34(2), 255-270.

[9] Lee, J.K.,Ryu, S.R. &Yoo K.Y. Fingerprint-based remote user authentication scheme using smart cards, Electronics Letters, 2002, 38(12), 554 – 555.

[10] Amina, R., Neeraj Kumara, Biswasb, G.P.,Iqbalc, R. & Victor Chang. A lightweight authentication protocol for IoT-enabled devices in distributed Cloud Computing environment, Future Generation Computer Systems, 2018, 78(3), 1005-1019.

[11] Gowthami, J & Shanthi, N. Secure Fuzzy Extractor based remote user validation scheme for Wearable devices. International Journal of Engineering research and Technology, RTICCT-2018, 6(8).

[12] Xue, K., Changsha Ma, Peilin Hong & Rong Ding, A Temporal-credential-based mutual authentication and key agreement scheme for wireless sensor networks, Journal of Network and Computer Applications, 2013, 36(1),316-323.

[13] Fatimah Al-alem, Mohammad, A. & Mahmoud Al-Ayyoub, On the road to the Internet of Biometric Things: A Survey of Fingerprint Acquisition Technologies and Fingerprint Databases, International Conference on Computer Systems and Applications, IEEE, 2016.

[14] Santoso, F.K. &Vun, N.C.H. Securing IoT for a smart home system. In International Symposium on Consumer Electronics (ISCE) 2015, Madrid, Spain, pp. 1–2.

[15] An, Y. Security analysis and enhancements of an effective biometric-based remote user authentication scheme using smart cards, J. Biomed. Biotechnology., 2012, pp. 1-6

[16] Das, A.K & Goswami A. A robust anonymous biometric-based remote user authentication scheme using smart cards.J King Saud Univ-Comput Inf Sci, 2014, 2(2), 193–210.

[17] Xinyi Huang, Yang Xiang, Ashley Chonka,Jianying Zhou &Robert H. Deng. A Generic Framework for Three-Factor Authentication: Preserving Security and Privacy in Distributed Systems. IEEE Transactions on Parallel and Distributed Systems, 2011,22(8).

[18] Manjulata, A.K. (2014). Survey on lightweight primitives and protocols for RFID in wireless sensor networks. Int J CommunNetw Inf Secur, 6(1), 29–40.

[19] Cremers C.J.F. (2008). The Scyther Tool: Verification, Falsification, and Analysis of Security Protocols. In: Gupta A., Malik S. (eds) Computer Aided Verification. Lecture Notes in Computer Science, 5123.

[20] Cremers, C.J.F. (2006). Scyther: semantics and verification of security protocols. Doi: 10.6100/IR614943.

[21] H. J. Kim and H. S. Kim, "AUTH HOTP - HOTP Based Authentication Scheme over Home Network Environment," in International Conference on Computational Science and Its Applications. Santander, Spain: Springer, 2011, pp. 622–637.D

[22] Sood, S.K., Sarje, A.K., Singh, K. (2011). A secure dynamic identity based authentication protocol for multiserver architecture. J Netw Comput Appl, 34(2), 609–618.

[23] Yoon, E., Yoo, K. (2013), Robust biometrics-based multi-server authentication with key agreement scheme for smart cards on elliptic curve cryptosystem. J Supercomput, 63(1),235–255

[24] Saru Kumari, Ashok Kumar Das, Xiong Li, Fan Wu, et al. (2017). A provably secure biometrics-based authenticated key agreement scheme for multi-server environments. Multimed Tools Appl, 77 (2), 2359–2389.

[25] B. Vaidya, J. H. Park, S. S. Yeo, and J. J. Rodrigues, "Robust onetime password authentication scheme using smart card for home network environment," Computer Communications, vol. 34, no. 3, pp. 326–336, 2011I

[26] Lee, C.C., Chen, C. T., Wu, P. H., & Chen, T. Y. (2013) Three-factor control protocol based on elliptic curve cryptosystem for universal serial bus mass storage devices, IET Computers & Digital Techniques, 7, 48–55.

[27] "Secure Hash Standard," FIPS PUB 180-1, National Institute of Standards and Technology (NIST), U.S. Department of Commerce, April1995. Available at http://csrc.nist.gov/publications/fips/fips180-4/fips-180-4.pdf. Last accessed on June 2018.

[28] Gowthami, J., Shanthi, N & Krishnamoorthy, N. Secure Three-factor remote user Authentication for E-Governance of Smart cities, International conference on Current Trends towards converging Technologies (ICCTCT), 2018. doi:10.1109/ICCTCT. 2018.8551172.

[29] Mittal, R.C. & Madhusudhan R.K. An enhanced biometrics-based remote user authentication scheme using mobile devices, International Journal of Computational Intelligence Studies, 2012,1(4). https://doi.org/10.1504/IJCISTUDIES.2012.050360.