Code Verification Work of Sybil Attack in Wireless Sensor Network

Gayatri Devi[1], Rajeeb Sankar Bal[2], Shubhashree Tripathy[3]

[1]Professor, Department of CSE, Ajay Binay Institute of Technology, Cuttack, Odisha, India
[2]Senior Lecturer, Department of CSE, Ajay Binay Institute of Technology, Cuttack, Odisha, India
[3]M.Tech (CSE) Student, Department of CSE, Ajay Binay Institute of Technology, Cuttack, Odisha, India

Abstract— *Wireless sensor networks are set to become a really pervasive technology that will influence our day by day life in imperative ways. WSNs undergo from many constraints, counting low computation capability, minute memory, restricted energy resources, susceptibility to physical capture, and the use of timid wireless communication channels. These constraints make security in WSNs a challenge. This section covers the different attacks and threats that relate to WSNs. A particularly harmful attack against sensor and ad hoc networks is known as the Sybil attack , where in a reputation system is subverted by forging identities in peer to peer network. In this paper, we propose Code verification technique to shield against the Sybil attack using a scheme namely Hwang et al.'s Scheme which is a password authentication scheme.*

Keywords— **Security, Sensor Network (SN), Wireless Sensor Network (WSN)**

I. INTRODUCTION

WSN are currently being employed in a variety of applications ranging from medical to military, and from home to industry. *WSN and Applications* aims to provide a reference tool for the increasing number of scientists who depend upon reliable SNs. Wireless sensors and WSN have come to the forefront of the scientific community recently. This is the consequence of engineering increasingly smaller sized devices, which enable many applications. The use of these sensors and the possibility of organizing them into networks have revealed many research issues and have highlighted new ways to cope with certain problems. In a typical scenario, users can retrieve information of interest from a WSN by injecting queries and gathering results from the so-called base stations (or sink nodes), which behave as an interface between users and the network. Thus, WSNs can be considered as a distributed database shown in figure -1.1.

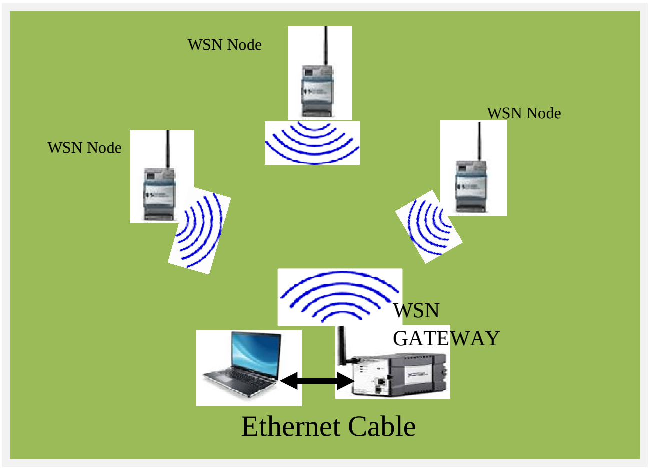


Fig-1.1(Wireless Sensor Network)

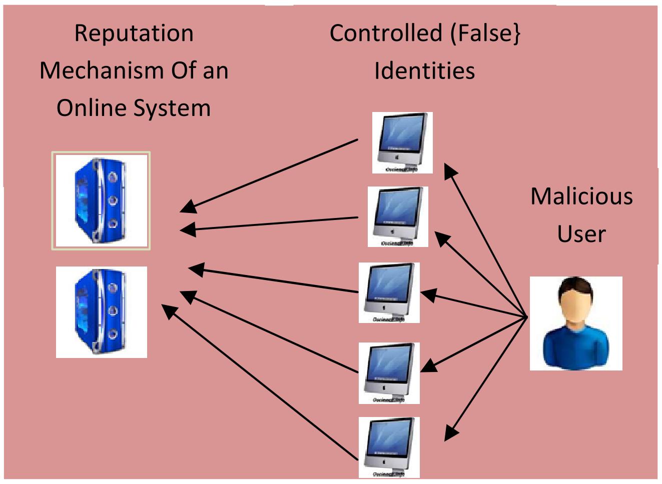

Reputation Mechanism Of an Online System
Controlled (False} Identities
Malicious User

Fig-1.2(Sybil Attack)

The Security in SN is complicated by the broadcast nature of the wireless communication and the lack of tamper-resistant hardware (to keep per-node costs low).
Additionally sensor nodes have restricted storage and computational resources rendering public key cryptography not viable. In this paper, we study the Sybil attack, a particularly harmful attack in sensor networks as

shown in figure -1.2.In the Sybil attack, a malicious node behaves as if it were a larger number of nodes, for example by claiming false identities. In the worst case, an attacker may generate an arbitrary number of additional node identities, using only one physical device. We propose several new defenses against the Sybil attack with key validation for random key predistribution, key verification, and registration during quantitative analysis, We also present a quantitative evaluation for the random key predistribution approach showing that it is robust to compromised nodes using Hwang et al.'s scheme.

II. SYBIL ATTACK

Sybil attack is a malicious device illegitimately taking on multiple identities. We refer, to malicious devices additional identities as Sybil nodes. This attack violates
One-to-one mapping by creating multiple identities [2] which is represented as shown in figure -1.2. To better understand the implications of the Sybil attack and how to defend against it, we develop a types of it as given below:

Types of Sybil Attack

In order to detect the Sybil attack it is necessary to understand the different forms in which the network is attacked [1].

(a) Direct and Indirect Communication
(b) Fabricated and stolen identities
(c) Simultaneous and non-simultaneous attack

Sybil Attack on Protocols

In a Sybil attack, a malicious node can generate and control a large number of identities on a single physical device. This gives the illusion to the network as if it were different legitimate nodes. It can affect the following important protocol [1]:

Distributed Storage: The Sybil attack affects the architecture where it replicates the data on several nodes. Data will be stored on Sybil identities. It means , While the system may be designed to replicate or fragment data across several nodes, it could actually be storing data on Sybil identities generated by the same malicious node.

Routing: Routing mechanism in which the nodes are supposed to be disjoint is affected by Sybil identities because one node will be present in the various paths and different locations at the same time.

Data Aggregation: In SN, data is grouped into one node to form complete information. When a Sybil node contributes many times posing as different users, the aggregated data changes completely thus giving false information. With enough Sybil nodes, an attacker may be able to completely alter the aggregate reading.

Voting: In WSN, most of the decisions are made by voting. Since the Sybil node has many identities, a single node has a chance of voting many times, thus destructing the process.

Misbehavior detection: A Sybil node increases the reputation, credit, trust value by using its virtual identities. Thus the accuracy to detect a malicious node is reduced. It is likely that any such misbehavior detector has some false positives. As a result, it might not take action until it observes several repeated offenses by the same node. An attacker with many Sybil nodes could "spread the blame", by not having any one Sybil identity misbehave enough for the system to take action.

Fair resource allocation: Since the Sybil node has multiple identities it affects the allocation of resources. For example, when many nodes share a single radio channel, each node will be assigned a fraction of time per interval during which they can transmit. Since the Sybil node has many identities, it can obtain an unfair share of the resources thus reducing the actual share of resources to the legitimate node.

Existing Detection Methods of Sybil Attack

(a) Radio resource testing: Consider that a node wants to verify that none of its neighbors are Sybil identities. It can assign each of its neighbors a different channel to broadcast some message on.[2] It can then choose a channel randomly on which to listen. If the neighbor that was assigned that channel is legitimate, it should hear the message. Let's' be the total number of the nodes 'n' be the number of Sybil nodes. The probability of detecting the Sybil node is s/n. A more difficult case is when there are not enough channels to assign each neighbor a different channel. In this case, a node can only test some subset of its neighbors at one time. If there are 'c' channels, then the node can test 'c' neighbors at once. Note that a malicious node not in the subset being tested can cover for a Sybil node that is being tested by transmitting on the channel that the Sybil node is supposed to be transmitting on.

(b) Registration: One obvious way to prevent the Sybil attack is to perform identity registration.[2] A difference between peer-to-peer networks and in WSN, there may be a trusted central authority managing the network, and thus knowing deployed nodes. The central authority may also be able to disseminate that information securely to the network. . To detect Sybil attacks, an entity could poll the network and compare the results to the known deployment. To prevent the Sybil attack, any node could check the list of "known-good'' identities to validate another node as legitimate. Registration is likely to be a good initial defense in many scenarios, with the drawbacks. The list of known identities must be protected from being maliciously modified. If the attacker is able to add identities to this list, he will be able to add Sybil nodes to the network.

(c) Position Verification: Another promising approach to defending against the Sybil attack is position verification. Here we assume that the sensor network is immobile once deployed. In this approach, the network verifies the physical position of each node. Sybil nodes can be detected using this approach because they will appear to be at exactly the same position as the malicious node that generates them [2]. By placing a limit on the density of the network, in-region verification can be used to tightly bind the number of Sybil identities that a malicious node can create.

***(d)* Random Key Predistribution:** Researchers recently proposed a promising technique for key distribution in sensor networks: random key predistribution [3, 4]. These techniques allow nodes to establish secure links to other nodes. In this section, we will show how these key distribution schemes can also be used to defend against the Sybil attack. In random key predistribution, we assign a random set of keys or key-related information to each sensor node, so that in the key set-up phase, each node can discover or compute the common keys it shares with its neighbors; the common keys will be used as a shared secret session key to ensure node-to-node secrecy.

Our key ideas are:

1. Associating the node identity with the keys assigned to the node.

2. Key validation, i.e., the network being able to verify part or all of the keys that an identity claims to have.

We use the following notation:

ID′: a randomly generated identity;

φ: key pool;

m: size of key pool, m = |φ|;

k: size of key ring

n: size of compromised key pool.

Using these we calculate the probability of SybilID as:

Pr(ID′ is a usable SybilID)=

$$\sum_{t=1}^{k}\left[\left(\binom{n}{t}\binom{m-n}{k-t}\right)/\binom{m}{k}\left(\frac{\binom{m-k+t}{k}}{\binom{m}{k}}\right)^{d}\right]$$

Using this formula we work here for code attestation (verification) work.

Code Attestation

Remote code verification or attestation is another promising new technique that could be employed to defend against many types of attacks, including the Sybil attack. The basic idea is to exploit the fact that the code running on a malicious node must be different from that on a legitimate node. Therefore, we could validate a node by verifying its memory content. So we applied here a New Scheme for Code Attestation & for indirect validation which effects on more security. We use here Hwang et al's Scheme which is derived from Lamport Algorithm[10]. In this, for proper and each sybilID a password is generated from server. Moreover this is a password authentication Scheme.

DEFENSES

To defend against the Sybil attack, we would like to validate that each node identity is the only identity presented by the corresponding physical node. There are two types of ways to validate an identity. The first type is direct validation, in which a node directly tests whether another node identity is valid. The second type is indirect validation, in which nodes that have already been verified are allowed to vouch for or refute other nodes. In this paper, several defenses are applied except in code verification work. For code verification, we use here Hwang et al.'s scheme [9] which is a password authentication scheme. For this we use random key predistribution [3,4,5,6] and key pool[7,8].

III. HWANG ET AL'S SCHEME

The notations used throughout this paper are described as in the following.

- U : a user.
- IDu, PWu: u's identifier & password respectively.
- S: remote server.
- h(.): Server hash function.
- ⊕ : bitwise XOR operation.
- n : a large prime number.
- g : a generator of Zn*.
- SK : S 's secret key.
- Vs : a solutions of the puzzle, decided by S .
- Ns : the nonce generated by S.
- Nu : the nonce generated by U.
- SKs : the secret key of S, used for puzzle verification.
- Tokenu : the message authentication code issued from S to U .
- Puzzle(P,x1,x2,..xn): Given p,x1,x2,…xn.
  find v such that h(x1,x2..xn, v) =P

Hwang et al.'s scheme involves three phases, the registration phase, the login phase and the verification phase, which can be described as in the following.

Registration phase:

In this phase, the user U initially registers with the server S.

1) U chooses his IDu and PWu , and sends them over a secure communication channel to S .

2) Upon receiving IDu and PWu , S computes

$$S_u=\sum_{t=1}^{k} \mathrm{IDu}^{SK}$$
$$h_u = g^{PWu.SK}$$

Login phase:
In this, user sends login request message to S.

$M1=\{\sum_{t=1}^{k} \mathrm{IDu}, \mathrm{Nu}\}$

Receiving M1, S checks. If already used, then rejects the session.
Otherwise, S determines Vs & generated Ns. Then computes :

$$P= h(\sum_{t=1}^{k} P, \mathrm{IDu}, N_u, N_s, v_s, SK)$$

$$token_u=h(\sum_{t=1}^{k} P, \mathrm{IDu}, \mathrm{Nu}, \mathrm{Ns}, \mathrm{Vs}, SK)$$

Then S generates M2 message as $M2=\{P, Ns, token_u\}$ & sends to U for login. Upon receiving M2 from server, It calculates:

$X_u=g^{ru.PWu}$ mod n

$Y_u=S_u \cdot hu^{r_u.token_u}$ *mod n*

Then user generates M3 message as $M3=(\sum_{t=1}^{k} IDu, PW, Tokenu, Xu, Yu)$ & sends to Server S.

Verification Phase:

Receiving M3 message S checks if $token_u$ equals $h(\sum_{t=1}^{k} P, \mathrm{IDu}, N_u, N_s, v_s, SK)$. If not stops the session.
Otherwise S verified as :

$$Yu^{SK^{-1}}=\mathrm{IDu}. X_u^{token_u}$$

Using these three phases, we work here for code verification.

IV. PROPOSED ALGORITHM

(A)For SybilID:

- define ID
- define m, n , k , t
- define d(Wheather for full validation or partial validation)
- compute probability of usable SybilID as :

Pr (ID′ is a usable SybilID) =

$$\sum_{t=1}^{k}[\,(\binom{n}{t}\binom{m-n}{k-t})\,/\binom{m}{k}\left(\frac{\binom{m-k+t}{k}}{\binom{m}{k}}\right)^{d}\,]$$

(B)For Registration Phase:

- define SK(secret key)
- define PW, S_u, h_u, g
- Then compute $S_u=\sum_{t=1}^{k} \mathrm{IDu}^{SK}$
- Compute $h_u=g^{PWu.SK}$

(C)For Login Phase:

- define N_u, N_s, v_s
- define r_u, X_u, Y_u
- compute p as $h(\sum_{t=1}^{k} \mathrm{IDu}, N_u, N_s, v_s)$ using appropriate hash function
- compute $token_u$ as $h(\sum_{t=1}^{k} P, \mathrm{IDu}, N_u, N_s, v_s, SK)$
- compute $X_u=g^{ru.PWu}$ mod n
- compute $Y_u=S_u\ .hu^{r_u.token_u}$ mod n

(C)For Verification Phase:

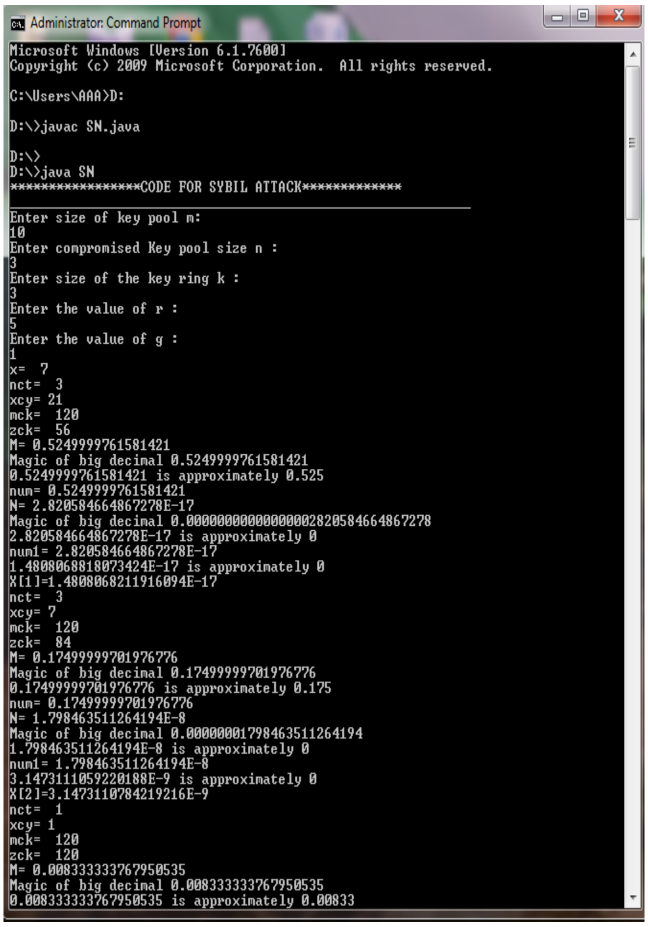
- compute $Yu^{SK^{-1}}=\mathrm{IDu}. X_u^{token_u}$
- compare and determine error difference

Following above algorithm, we developed a java code with assuming small number taken to the variables. The example is given below. In this there is full validation phase and partial validation phase occurred which is determined by d. For full validation d=50 and for partial validation d=30 is taken.

V. RESULT ANALYSIS

```
****CODE FOR SYBIL ATTACK**********
_______________________________________________
Enter size of key pool m:
10
Enter compromised Key pool size n :
3
Enter size of the key ring k :
3
Enter the value of r :
5
Enter the value of g :
1
x=  7
nct=  3
xcy= 21
mck=  120
zck=  56
M= 0.5249999761581421
Magic of big decimal 0.5249999761581421
0.5249999761581421 is approximately 0.525
num= 0.5249999761581421
N= 2.820584664867278E-17
Magic of big decimal
0.00000000000000002820584664867278
2.820584664867278E-17 is approximately 0
num1= 2.820584664867278E-17
1.4808068818073424E-17 is approximately 0
X[1]=1.4808068211916094E-17
nct=  3
xcy= 7
```

```
mck=  120
zck=  84
M= 0.17499999701976776
Magic of big decimal 0.17499999701976776
0.17499999701976776 is approximately 0.175
num= 0.17499999701976776
N= 1.798463511264194E-8
Magic of big decimal 0.00000001798463511264194
1.798463511264194E-8 is approximately 0
num1= 1.798463511264194E-8
3.1473111059220188E-9 is approximately 0
X[2]=3.1473110784219216E-9
nct=  1
xcy= 1
mck=  120
zck=  120
M= 0.008333333767950535
Magic of big decimal 0.008333333767950535
0.008333333767950535 is approximately 0.00833
num= 0.008333333767950535
N= 1.0
Magic of big decimal 1.0
1.0 is approximately 1
num1= 1.0
0.008333336915261641 is approximately 0.00833
X[3]=0.008333336561918259
Magic of big decimal
0.000000000000000014808068211916094
ID[1]=0
SUM2 =1.4808068211916094E-17
TOTAL SUM =2.961613642383219E-17
SK=0.3952986360030897
PW=0.4533619606091347
S[1]=0
Magic of big decimal
0.0000000031473110784219216
ID[2]=0
SUM2 =3.1473110784219216E-9
TOTAL SUM =6.294622156843843E-9
SK=0.8346470939978052
PW=0.7816371828920409
S[2]=0
Magic of big decimal 0.008333336561918259
ID[3]=0.00833
SUM2 =0.008333336561918259
TOTAL SUM =0.016666673123836517
SK=0.1489686368763865
PW=0.817614501037797
S[3]=0.49008
.............REGISTRATION PHASE.........
User ID is :0.49008
v[1]=4
v[2]=6
v[3]=8
............LOGIN PHASE...........
p= 9
SK=0.44180605719302135
j[1]=28
SK=0.6244802221402853
j[2]=30
SK=0.6160265039783728
j[3]=32
token=9
X= 1
h = 1
Y= 0
........VERIFICATION PHASE........
Inverse of Secret Key Is :1.2938430960111162
o1=0.0
o2=0.016666673123836517
Error Difference of above given value is :0.016666673123836517
```

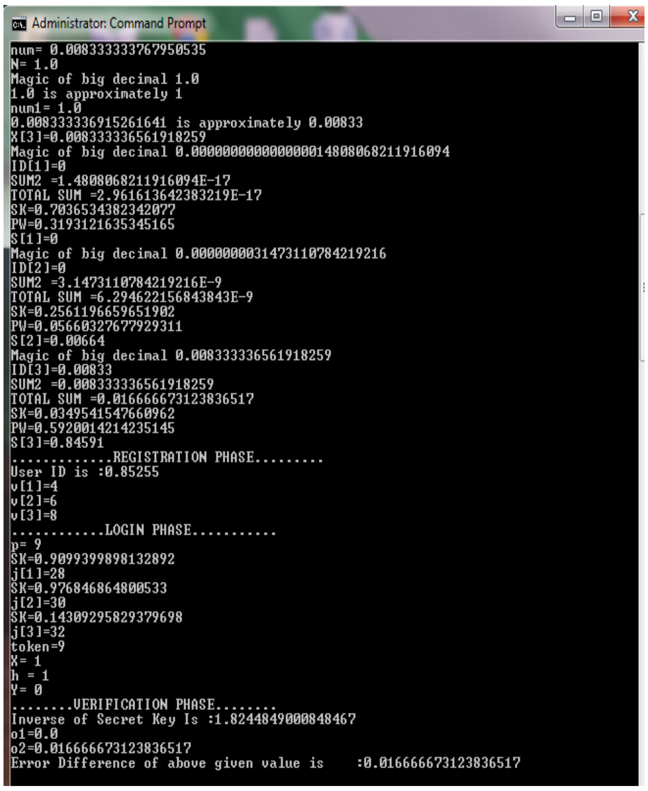


```
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\AAA>D:

D:\>javac SN.java

D:\>
D:\>java SN
*****************CODE FOR SYBIL ATTACK*************
__________________________________________________________
Enter size of key pool m:
10
Enter compromised Key pool size n :
3
Enter size of the key ring k :
3
Enter the value of r :
5
Enter the value of g :
1
x=  7
nct=  3
xcy= 21
mck=  120
zck=  56
M= 0.5249999761581421
Magic of big decimal 0.5249999761581421
0.5249999761581421 is approximately 0.525
num= 0.5249999761581421
N= 2.820584664867278E-17
Magic of big decimal 0.00000000000000002820584664867278
2.820584664867278E-17 is approximately 0
num1= 2.820584664867278E-17
1.4808068818073424E-17 is approximately 0
X[1]=1.4808068211916094E-17
nct=  3
xcy= 7
mck=  120
zck=  84
M= 0.17499999701976776
Magic of big decimal 0.17499999701976776
0.17499999701976776 is approximately 0.175
num= 0.17499999701976776
N= 1.798463511264194E-8
Magic of big decimal 0.00000001798463511264194
1.798463511264194E-8 is approximately 0
num1= 1.798463511264194E-8
3.1473111059220188E-9 is approximately 0
X[2]=3.1473110784219216E-9
nct=  1
xcy= 1
mck=  120
zck=  120
M= 0.008333333767950535
Magic of big decimal 0.008333333767950535
0.008333333767950535 is approximately 0.00833
```



```
num= 0.008333333767950535
N= 1.0
Magic of big decimal 1.0
1.0 is approximately 1
num1= 1.0
0.008333336915261641 is approximately 0.00833
X[3]=0.008333336561918259
Magic of big decimal 0.000000000000000014808068211916094
ID[1]=0
SUM2 =1.4808068211916094E-17
TOTAL SUM =2.961613642383219E-17
SK=0.7036534382342077
PW=0.3193121635345165
S[1]=0
Magic of big decimal 0.0000000031473110784219216
ID[2]=0
SUM2 =3.1473110784219216E-9
TOTAL SUM =6.294622156843843E-9
SK=0.2561196659651902
PW=0.05660327677929311
S[2]=0.00664
Magic of big decimal 0.008333336561918259
ID[3]=0.00833
SUM2 =0.008333336561918259
TOTAL SUM =0.016666673123836517
SK=0.0349541547660962
PW=0.5920014214235145
S[3]=0.84591
.............REGISTRATION PHASE.........
User ID is :0.85255
v[1]=4
v[2]=6
v[3]=8
............LOGIN PHASE...........
p= 9
SK=0.9099399898132892
j[1]=28
SK=0.976846864800533
j[2]=30
SK=0.14309295829379698
j[3]=32
token=9
X= 1
h = 1
Y= 0
........VERIFICATION PHASE........
Inverse of Secret Key Is :1.8244849000848467
o1=0.0
o2=0.016666673123836517
Error Difference of above given value is    :0.016666673123836517
```

V. CONCLUSION

In this paper, we define the Sybil attack and types of this attack. Hwang et al.'s Scheme is used for code verification work. A particular java code is developed for this purpose for both full validation (d=50) and partial validation (d=30). This code gives best result on m=10 and n<=5. A little error difference is occurred. In this scheme Secret Key and Password is given by the user which is insecure. For this only, we use here random numbers which is in between 0 to 1 for both Password and Secret key which gives better security.

This code is valid upto m=30, Above m=30, this code will stop because of receiving huge amount of fractions. Implement this in other tool like NS2 will give better result. For more security we can use dynamic verification like CAPTCHA in code verification work which is left for future purpose.

VI. REFERENCES

[1] J. Newsome, E. Shi, D. Song and A. Perrig:"The Sybil Attack in Sensor Network: Analysis & Defences," The Third Intl. Symposium on Information Processing in Sensor Networks (IPSN'04), Berkeley, California, USA: ACN Press, 2004, pp.185-191.

[2] J.R. Douceur. The Sybil attack. In First International Workshop on Peer-to Peer Systems (IPTPS'02), Mar. 2002.

[3] H. Chan, A. Perrig, and D. Song. Random key predistribution schemes for sensor networks. In IEEE Symposium on Security and Privacy, May 2003.

[4] W. Du, J. Deng, Y. S. Han, and P. K. Varshney. A pairwise key pre-distribution scheme for wireless sensor networks. In ACM CCS 2003, pages 42–51,Oct. 2003.

[5] L. Eschenauer and V. D. Gligor. A key-management scheme for distributed sensor networks. In Proceedings of the 9th ACM Conference on Computer and Communication Security, pages 41–47, Nov. 2002.

[6] D. Liu and P. Ning. Establishing pairwise keys in distributed sensor networks. In ACM CCS 2003, pages 52–61, Oct. 2003.

[7] Next-Generation Secure Computing Base (NGSCB). http://www.microsoft.com/resources/ngscb/ default.mspx, 2003.

[8] R. D. Pietro, L. V. Mancini, and A. Mei. Random key assignment for secure wireless sensor networks. In ACM Workshop on Security of Ad Hoc and Sensor Networks, 2003.

[9] Hwang M.-S., Chong S.-K., Chen T.-Y., DoS-resistant ID-based password authentication scheme using smart cards, The Journal of Systems and Software 83 (2010) 163–172.
[10] Lamport L., Password authentication with insecure communication, Communications of ACM 24 (1981) 770–772.