INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Authentication and Proofing Using Zero Knowledge Protocol

T.S.Brinda[1], R.Ranjeetha[2], M.Yesodha[3], V.Manjuarasi4

Assistant Professor, Dept of ECE, Prince Dr. K. Vasudevan College of Engineering and Technology, TamilNadu, India

Abstract: *Wireless sensor network (WSN) have become a prominent solution for various interesting applications like security surveillance and monitoring of geographical areas. Various types of sensors are used and deployed in the network to collect useful physical parameters and some highly sensitive information is then been transmitted between the nodes and to the base station, without any human intervention. Hence, message authenticity and security are major requirements in WSN. Since the cryptographic schemes used for wired networks tend to exhaust wireless sensor network resources, they cannot be directly used in sensor networks. In this, the zero knowledge protocol (ZKP) is investigated that it is an under applied authentication mechanism used to identify compromised nodes from genuine ones. Here the Zero Knowledge Protocol (ZKP) is implemented in the network for the authentication and verification of sender sensor nodes before transmitting any sensitive information. The proposed scheme addresses the improvement in the security while maintaining the message confidentiality. In proposed scheme an optimal number of challenge questions are also used to maintain a balance between the added security and the increase in cost. Increase in the number of challenge question makes up to the reduced key size thus providing an improved security. The proposed scheme was assessed based on the mat lab simulation and an analysis was performed.*
Keywords: *Zero Knowledge Protocol, Sensor nodes, Cryptography, Prover, Verifier*

I. INTRODUCTION

Sensor Nodes are basically battery powered with limited computation capability, Cluster Heads are Nodes that are more powerful than sensor Nodes and Base Stations are resource abundant. The key job of Nodes is to continuously collect data for events of interests and deliver the data to a designated Cluster Head. The major job of Cluster Head is to aggregate all the data's received and to send to a Base Station. Implementing these authentication mechanisms becomes very difficult because of their design constrains, due to limited resources and their physically insecure nature. The network autonomously must be able to identify and prevent itself from these attacks. Due to an insecure physical hardware and light weight operating systems, the network is prone to clone attack. In case of WSNs it is very easy for an adversary to capture or clone nodes and place them into the network by copying the cryptographic information. Malicious packet injection is also very common through man in the middle attack. Few of the highly regarded cryptographic mechanism such as RSA used in wired networks which proposed solutions for the above mentioned attacks cannot be used in WSNs because of the lack of memory and computation power and constrains on energy consumption, making it inappropriate.

II. ZERO KNOWLEDGE PROTOCOL

In cryptography, a zero-knowledge protocol is a method by which one party (the prover) can prove to another party (the verifier) that a given statement is true, without conveying any additional information apart from the fact that the statement is indeed true. Notice that the notion only applies if the statement being proven is the fact that the prover has such knowledge. This is a particular case known as zero-knowledge proof of knowledge, and it nicely illustrates the essence of the notion of zero-knowledge proofs: proving that one possesses a certain knowledge is in most cases trivial if one is allowed to simply reveal that knowledge; the challenge is proving that one has such knowledge without revealing it or without revealing anything else.For zero-knowledge proofs of knowledge, the protocol must necessarily require interactive input from the verifier, usually in the form of a challenge or challenges such that the responses from the prover will convince the verifier if and only if the statement is true (i.e., if the prover does have the claimed knowledge). This is clearly the case, since otherwise the verifier could record the execution of the protocol and replay it to someone else: if this were accepted by the new party as proof that the replaying party knows the secret information, then the new party's acceptance is either justified - the replier 'does' know the secret information - which means that the protocol leaks knowledge and is not zero-knowledge, or it is spurious. Research in zero-knowledge proofs has been motivated by authentication systems where one party wants to prove its identity to a second party via some secret information (such as a password) but doesn't want the second party to learn anything about this secret. This is called a "zero-knowledge proof of knowledge". However, a password is insufficiently random to be used in many schemes for zero-knowledge proofs of knowledge.

A. Authentication using Zero Knowledge Protocol

The zero knowledge proof relies on the fact that during the entire process of authentication the secret key or password is not revealed to the receiver node. The nodes receive information only after successful authentication from the receiver node. There is a

zero knowledge protocol mechanism between each node and cluster heads. Hence there is an authentication process before any transmission of data takes place.

The entire process of authentication (fig 1) can be explained in the following steps:

Step1: The prover P chooses a random number r, calculates x (eq. 3)

Step2: The prover P then send x to the verifier

Step3: Now the verifier requests for the prover's protocol key vp from the base station accompanied by its own protocol key $vv = sv^2 \bmod N$ where sv is the secret key of the verifier.

Step4: Now the base station calculates the protocol key of the verifier $z = sv^2 \bmod N$, using the secret key stored for corresponding node id.

Step5: The base station then compares value z and protocol key received from the verifier. If both keys are equal, it authenticates the verifier

Step6: After successful authentication the base station computes the protocol key of the prover and sends it to the verifier. Transferring of protocol key in a multi-hop network will not affect his security as an adversary cannot deduce secret key from the protocol key.

Step7: The verifier V now chooses a random challenge question e (e=0, 1, 2, 3, 4), asks and ask the prover P for $Y = rs^e \bmod N$.

Step8: Prover then calculates Y using random number r, secret key s and the challenge question e.

Step9: Prover then sends back Y as a response to the challenge question.

Step10: the verifier then compute and two values $val1 = Y^2 \bmod N$ and $val2 = xv^e \bmod N$ with each other. Val1 and val2 will only be equal if the secret provided by the prover matches the secret key provided by the base station. As BS is the trusted party, the key match will prove the authenticit of thprover P.
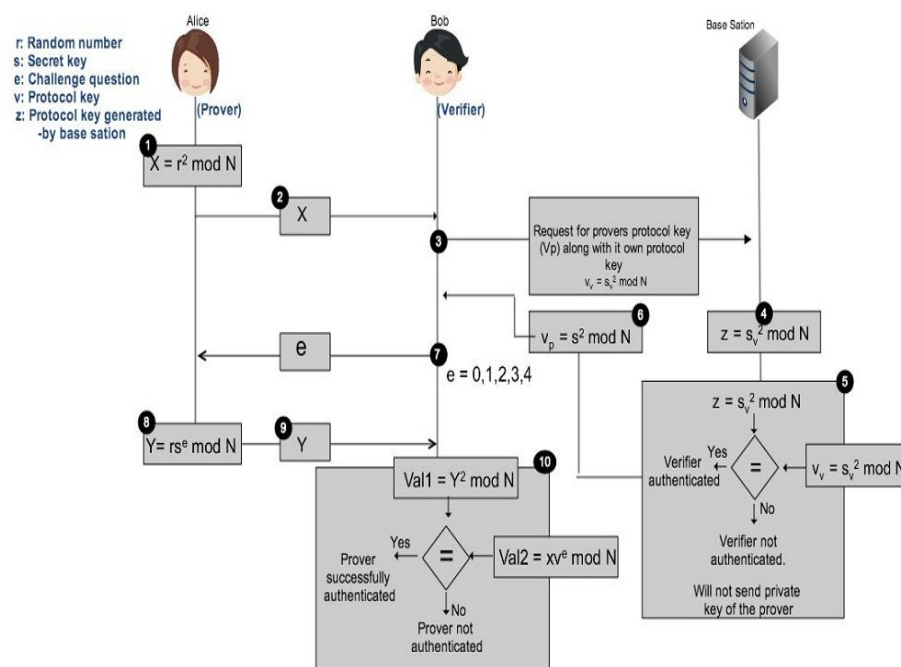

Fig.1 Authentication Mechanism

The authentication process is initialized by the prover by generating a random number r and calculates $x = r^2 \bmod N$

$$x = r^2 \bmod N \qquad (1)$$

Prover then sends x to the verifier (Node 4). Now the verifier will request the base station for the protocol key v_2 (eq. 2)of the prover and sends its own protocol key $v_4 = s_4^2 \bmod N$ along with it.

$$V_2 = s_2^2 \bmod N \qquad (2)$$

The base station will first compute the protocol key of the verifier using the stored secret key of the same (s_4) and then compares it with the protocol key received from the verifier. If both values are found equal, then the BS replies back with the protocol key (v_2) of the prover. This mechanism is termed as two-way authentication in our proposed method. This mechanism is implemented to authenticate the verifier node before the base station shares any protocol key, hence improving security in the network. The verifier will ask a random challenge question to the prover after receiving the protocol key of the prover from the base station. This

challenge question will be a random value of e. Based upon the challenge question asked by the verifier, the prover replies with Y (eq. 3). The verifier then calculates and compares two values val1 (eq. 4) and val2 (eq. 6).

$$Y = rs_2{}^e modN \qquad (3)$$
$$val1 = Y^2 modN \qquad (4)$$
$$val1 = Y^2 modN = (rs_2{}^e)^2 modN = r^2 s_2{}^{2e} modN \qquad (5)$$
$$val2 = xv_2{}^e modN \qquad (6)$$
$$val2 = xv_2{}^e modN = r^2 modN \qquad (7)$$
$$(s_2{}^2)^e modN = r^2 s_2{}^{2e} modN \qquad (8)$$

For example for challenge question e=1:

$$val1 = r^2 s_2{}^2 modN \qquad (9)$$
$$val2 = r^2 s_2{}^2 modN \qquad (10)$$

The secret key s_2 in (eq. 4) comes from the prover and the secret key s_2 in (eq. 6) comes from the base station. This way both val1 and val2 will only be equal if both the secret keys match with each other i.e. if the prover is legitimate and is using the secret key stored during the pre-deployment stage. By using this protocol the verifier can compare and match the secret key of the prover from two sources without even learning the actual value of the secret key of the prover.

This complete process of authentication, except for protocol key exchange, is repeated K times and for each round, a new random number rand random challenge question e is chosen. The protocol also requires the response to a challenge to be provided within a time limit such that it becomes computationally infeasible for an impersonator to answer to the challenge by using brute force method. This authentication mechanism is performed before every initialization of data transmission. For every authentication process, a new public key N is generated by the base station. The entire process of authentication is shown in fig.2, where Alice acts as the prover and Bob acts as verifier and public key N is shared among them by the base station.

In ZKP, the challenge question e plays a very crucial part to authenticate the sender node. If an adversary in the beginning of the authentication process possesses the challenge question e, it can easily prove itself to be a genuine sender. Hence it is very important that the authentication process is repeated multiple times and challenge key e should be chosen randomly for each round.
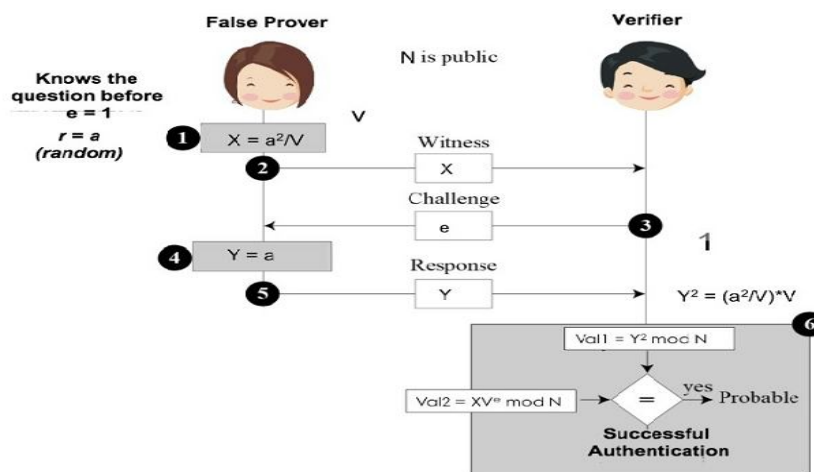

Fig.2 False claimant authentication in ZKP

For an instance, if a false claimant knows the challenge question e before the start of the authentication process, it can claim to be a genuine node by generating an arbitrary number a and sends $x = a^2/v^e modN$ to the verifier. Upon receiving the expected value of e, the false claimant sends Y = a. The verifier will then computes $val1 = Y^2 modN$, which becomes $a^2 modN$. It also computes $val2 = xv^e modN$ which becomes $a^2/v^e modN$. This process will result in val1 and val2 to be equal. This will lead to false authentication. In [2] only two challenge questions were used (e = 0, 1).

Here the number of challenge questions increased to 5 (e = 0, 1, 2, 3, 4). By having more options for e value we have increased security of the protocol. If we have only one option for e i.e. e=0, then at all rounds the adversary can authenticate itself and ZKP will fail. If we have two options for e, i.e. e=0 and e=1, we will have a good security but even then the randomness can be predicted by high probability. So if we have more options for e, the security is increased y several folds, but simultaneously the computation cost will also increase. So through our simulation we found an optimal number of challenge questions to maintain a balance between the number of challenge questions and that of the computation cost. By increasing the number of challenge questions we were able to reduce the number of rounds the authentication must be performed.

In [2] when e has 2 values the optimal number of rounds was 10, which makes the probability of successful authentication by a malicious node to be $(1/2)^{10}$. Here ZKP able to improve the security with lesser number of rounds (K), by increasing the number of

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor:6.887
Volume 5 Issue IX, September 2017- Available at www.ijraset.com

challenge questions e. After successful authentication of the node K times, the data transfer phase starts, wherein the nodes exchange information.

III.RESULTS & DISCUSSION

Here a model of ZKP scheme is developed in MATLAB, and various performance analyses were done.

A. Computation cost

Computation cost for proposed protocol can be derived as (eq. 11), where t is the number of rounds, l being number of challenge questions and k is the multiplicity of challenge. The simulation results of the proposed network model and its computation results are shown below.

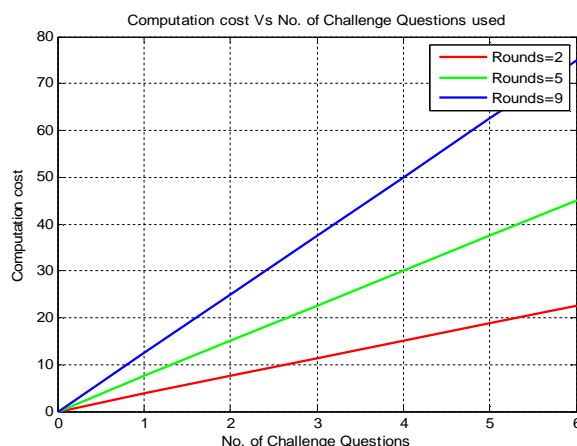$$\text{Computaton cost} = t.l.(k + 1)/4 \tag{11}$$

Fig.3.Number of challenge questions versus the computation plot

In Fig.3, we can notice that as number of challenge question increases, the computation cost also increases. We can also see that computational cost will increase more rapidly with the increase in number of rounds.

B. Security

In fig.4, ZKP performs better when numbers of challenge questions are increased. Security in the system is measured with increase in challenge questions and we found that with increase in number of challenge questions, lesser number of rounds are required to provide optimum security than in [2].
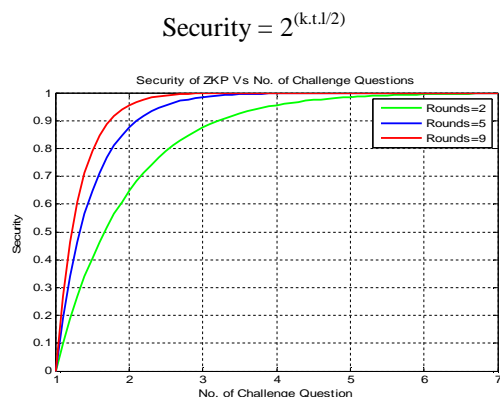
$$\text{Security} = 2^{(k.t.l/2)}$$

Fig.4. Security of the proposed ZKP model.

REFERENCES

[1] K.R. Venugopal and L.M. Patnaik, "Authentication in WirelessSensor Networks Using Zero Knowledge Protocol", ICIP 2011,CCIS 157, pp. 416¸S421, 2011.Springer-Verlag Berlin Heidelberg 2011

[2] [2] Siba K. Udgata, AlefiahMubeen, Samrat L. Sabat, "WirelessSensor Network Security model using Zero Knowledge Protocol" Proceedings of ICC, IEEE International Conference, 2011,pages 1-5

[3]   Limin Ma, Yu Ge, Yuesheng Zhu, "TinyZKP: A Lightweight Authentication Scheme Based on Zero-Knowledge Proof for Wireless Body Area", Wireless PersCommun DOI 10.1007/s11277-013-1555-4, Springer Science Business Media New York 2013

[4]   Kai Xing Fang, Liu Xiuzhen, Cheng David, H. C. Du, "Real-time Detection of Clone Attacks in Wireless Sensor Networks ",Proceedings of the 28th International Conference on DistributedComputing Systems, 2008, Pages 3-10

[5]   Nikos Komninos, DimitrisVergados, Christos Douligeris, "Detecting Unauthorized and Compromised Nodes in Mobi Adhoc" Networks Journal of Ad Hoc Networks, Volume 5, Issue3, April 2007, Pages: 289-298

[6]   KlempousRyszard, Nikodem Jan, Radosz Lukasz, Raus Norbert, "Adaptive Misbehaviour Detection in Wireless Sensors Network Based on Local Community Agreement", 14th Annual IEEE International Conference and Workshops on the Engineering of Computer Based systems, ECBS 2007, 2007,Page(s):153-160

[7]   KrontirisIoannis, TassosDimitriou and Felix C. Freiling,"Towards Intrusion detection In Wireless Sensor Networks", InProc. of the 13th European Wireless Conference, 2007

[8]   Joseph Binder, Hans Peter Bischof, "Zero Knowledge Proofs ofIdentity for Ad Hoc Wireless Networks An In-Depth Study",Technical Report, 2003. http://www.cs.rit.edu/ jsb7384/zkpsurvey.pd

[9]   Goldwasser, S., Micali, S., Rackoff, " The Knowledge Complexity of Interactive Proof Systems". SIAM J. Computing 18,186â A¸S208 (1989)

[10] Feige, U., Fiat, A., Shamir, "A.: Zero Knowledge Proofs ofIdentity". J. Cryptology 1, 77âS94 (1988)

[11] H.Choi, S.Zhu, and T.Laporta., "Set: Detecting Node Clones inSensor Networks". InSecureCommâZ07, 2007.

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)