Journal of Science and Technology (JST)
Volume 2, Issue 4, July - August 2017 PP 07-16 *ISSN:2456- 5660*

ESPK: External agent authentication and session key establishment using public key technique in wireless sensor networks

Swasthika Jain T.J [1], Dr Brahmananda S H [2]
[1](Department of CSE, GITAM School of Technology ,Bengaluru, India)
[2](Department of CSE, GITAM School of Technology ,Bengaluru, India)

Abstract : *Wireless sensor networks (WSNs) have recently attracted a lot of interest in the research community due their wide range of applications. Due to distributed and deployed in a un attend environment, these are vulnerable to numerous security threats. In this paper, describe the design and implementation of public-key-(PK)-based protocols that allow authentication and session key establishment between a sensor network and a third party. WSN have limitations on computational capacity, battery etc which provides scope for challenging problems. We fundamentally focused on the security issue of WSNs The proposed protocol is efficient and secure in compared to other public key based protocols in WSNs.*

Keywords - *Authentication, Wireless sensor network, Cryptography key management, Encryption, public key techniques. Rivest Shamir Adelman (RSA).*

I. INTRODUCTION

Wireless sensor networks (WSNs) consist of hundreds or even thousands of small devices each with sensing, processing, and communication capabilities to monitor the real-world environment such as light, heat, pressure etc. Wireless sensor networks have been proposed for such applications include Battle ground surveillance, Enemy movement, Environmental monitoring, Forest fire monitoring, Elephant monitoring in villages, Tracking patients, doctors, drug administrators etc. reporting data to end users, habitat monitoring [2], structural health monitoring [3], emergency medical care [4], and vehicular tracking [5] all of which demand some combination of authentication, integrity, privacy and security. Sensor devices are limited in their energy, computation and communication capabilities. A typical node (Berkeley mote)[4]in WSN have a configuration of 8-bit CPU (4MHz),128KB flash, 4KB RAM and Transmission range of 100 Feet. Sensor networks closely interact with their physical environments and with people, posing new security problems. Security in wireless sensor network is different compared to security in LANS, WANS, Internet etc. In WSNs one of the primary security requirements is authentication of entity, message, data, especially in data critical applications. Most of the protocols [6], [9] are based on Symmetric key cryptography. Even though Symmetric cryptography algorithms are typically fast and are suitable for processing large streams of data, it presumes two parties have agreed on a key and been able to exchange that key in a secure manner prior to communication. The agreement of both the parties to use the same private key before they actually start the communication poses lots of problem. Most of the algorithms focus on communication between the sensor nodes or between the sensor node and an external agent. The communication between sensor and External agent is not dealt appropriately. The implementation of RSA and ECC cryptography on Mica2 [7] nodes further proved that a public key based protocol is viable for WSNs. In [1], Watro et al have described a system names TinyPK where RSA system has been implemented using Public key cryptography. Public key (PK) technology is a widely used tool to support symmetric key management in the realm of Internet hosts and high-bandwidth interconnections. The Public key cryptography is possible in sensor nodes, by selecting appropriate parameters, for example using the small integer e = 3 as the public key [8][10][12], the public key operation can be extremely fast. The base station and external party have enough computational resources in terms of CPU, Memory, and Battery and based on Watro et al work, we proposed a protocol which is highly secure and used for Authentication and key establishment between sensor and external agent.

II. CONSTRAINTS IN WIRELESS SENSOR NETWORKS

A wireless sensor network consists of a large number of sensor nodes which are inherently resource-constrained. These nodes have limited processing capability, very low storage capacity, and constrained communication bandwidth. These limitations are due to limited energy and physical size of the sensor nodes. Due to these constraints, it is difficult to directly employ the conventional security mechanisms in WSNs. In order to optimize the conventional security algorithms for WSNs, it is necessary to be aware about the constraints of sensor nodes [13]. Some of the major constraints of a WSN are listed below.

*Energy constraints***:** Energy is the biggest constraint for a WSN. In general, energy consumption in sensor nodes can be categorized in three parts: (i) energy for the sensor transducer, (ii) energy for communication among sensor nodes, and (iii) energy for microprocessor computation. The study in [14] found that each bit transmitted in WSNs consumes about as much power as executing 800 to 1000 instructions. Thus, communication is more costly than computation in WSNs. Further, higher security levels in WSNs usually correspond to more energy consumption for cryptographic functions.

Memory limitations: A sensor is a tiny device with only a small amount of memory and storage space. Memory is a sensor node usually includes flash memory and RAM. Flash memory is used for storing downloaded application code and RAM is used for storing application programs, sensor data, and intermediate results of computations.

Unreliable communication: Unreliable communication is another serious threat to sensor security. Normally the packet-based routing of sensor networks is based on connectionless protocols and thus inherently unreliable. Packets may get damaged due to channel errors or may get dropped at highly congested nodes. Furthermore, the unreliable wireless communication channel may also lead to damaged or corrupted packets. This is due to the broadcast nature of wireless communication, as the packets may collide in transit and may need retransmission [11].

Unattended operation of networks: In most cases, the nodes in a WSN are deployed in remote regions and are left unattended. Remote management of a WSN makes it virtually impossible to detect physical tampering. This makes security in WSNs a particularly difficult task.

III. SECURITY MECHANISMS FOR WSNS

In this section, defense mechanism for combating various types of attacks on WSNs will be discussed. First, different cryptographic mechanisms for WSNs are presented. Both public key cryptography and symmetric key cryptographic techniques are discussed for WSN security. A number of key management protocols for WSNs are discussed next. Various methods of defending against DoS attacks, secure broadcasting mechanisms and various secure routing mechanisms are also discussed. In addition, various mechanisms for defending the Sybil attack, node replication attack, traffic analysis attacks, and attacks on sensor privacy are also presented. Finally, intrusion detection mechanisms for WSNs, secure data aggregation mechanisms and various trust management schemes for WSN security are discussed.

*A) Cryptography In WSNs :*Selecting the most appropriate cryptographic method is vital in WSNs as all security services are ensured by cryptography. Cryptographic methods used in WSNs should meet the constraints of sensor nodes and be evaluated by code size, data size, processing time, and power consumption. In this section, we focus on the selection of cryptography in WSNs. We discuss public key cryptography first, followed by symmetric key cryptography.

Symmetric key cryptography in WSNs

Since most of the public key cryptographic mechanisms are computationally intensive, most of the research studies for WSNs focus on use of symmetric key cryptographic techniques. Symmetric key cryptographic mechanisms use a single shared key between the two communicating host which is used both for encryption and decryption.

B) Key Management Protocols: The area that has received maximum attention of the researchers in WSN security is key management. Key management is a core mechanism to ensure security in network services

and applications in WSNs. The goal of key management is to establish the keys among the nodes in a secure and reliable manner. In addition, the key management scheme must support node addition and revocation in the network. Since the nodes in a WSN have computational and power constraints, the key management protocols for these networks must be extremely light-weight.

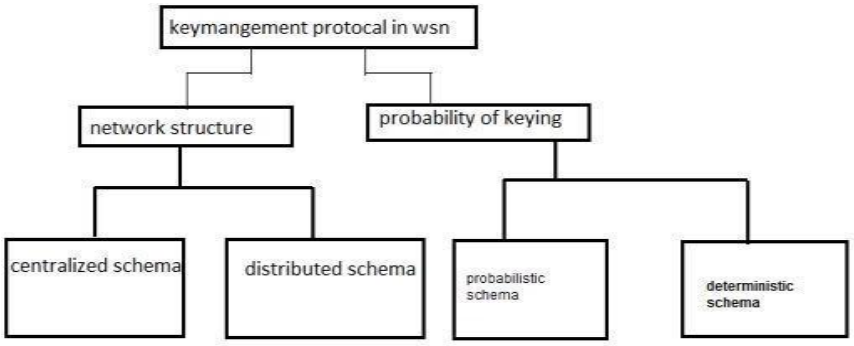

Figure 1 A taxonomy of key management protocols in WSNs

IV. PROTOCOL BASED ON RSA

*Tiny PK***:** Tiny PK security scheme is proposed by Watro et al [1]. The security scheme is used to authenticate the user (external agent) and allows the sensor to share a session key with external agent. The infrastructure requirement for TinyPK is CA, EA and WSN.CA is a trusted Certification Authority, which is an entity with public and private keys.CA is a trusted entity by all friendly units. EA is an External agent is an entity who tries to communicate with a sensor of WSN. Every mote is loaded with CA public key while deploying into the network. Based on the figure provided below, we depict the functioning of Tiny PK and in next section we discuss the limitations of Tiny PK. The Tiny PK is Challenge- Response security mechanism in which the external agent sends a challenge to the Sensor network. Based on the challenge the WSN authenticates the external agent and allows the sensor to establish a session key with external agent. The challenge consists of the public key of External agent which is encrypted with the private key of CA i.e. Ciphertext1= $E_{CAPvtke\ y}$ $\{EPubKey\}$ which provides Authentication to WSN. Challenge also contains a nonce (a time stamp, used to detect replay attacks) and checksum of the public key of external agent forms Cipherext2, encrypted with the private key of external agent i.e. Ciphertext2= $E_{EPVTkey}$ {nonce||chk} where chk=h (EPubKey), which provides security on transmission through an insecure public channel. Check sum is used to check the integrity of external agent public key. Once the challenge is received by a sensor, it decrypts the Ciphetext1 with the already pre-loaded public key of CA i.e.$D_{CApubkey}$ {Ciphertext1} = EPubKey, by using the public key of external agent, sensor decrypts Ciphertext2 i.e. $D_{Epubkey}$ {Ciphertext2} to get nonce and check sum. Sensor validates the check sum and nonce, if everything is valid then sensor generates a Session key called TinySessionKey and encrypts with public key of external agent i.e. Ciphertext3= $E_{EPubkey}$ {TinySessionKey||nonce}. On receiving of response from sensor, the EA decrypts the message with its private key i.e. $D_{Epvtkey}$ {Ciphertext3} = {TinySessionKey||nonce} and verifies nonce.

Limitations of Tinypk

The main aim of public key cryptography is that the public keys of entities must be made public and private keys must be kept secret. In the above algorithm given by Watro [1], the public key CA and External Agent (EA) is not made public, which adds complexity in the protocol to send public key of External agent to sensor.

• If an intruder come to know the public key of CA which is made public then he can forge the system.

• Assume an Adversary knows the public key of CA. He can decrypt the Ciphetetx1 which gives him the public key of EA. from the EA public key; he decrypts the Ciphetext2 through which he gets the nonce and checksum; through checksum he can validate the EA public key. Now he can frame a response to EA through nonce, and a session key which is Ciphettext3. On receiving of the response from intruder the EA checks the nonce and validates and once it is valid, it comes into session with adversary, assuming it a sensor. Once Session starts the adversary can forge the WSN.

• The TinyPK is suffering the task of loading of each sensor with the public key of CA. which is time consuming and inefficient. As the public key must be available public in PBK cryptography.

• Once challenge is received by WSN, then any of the sensor can respond as every node contains the public key of CA. which causes node colluding. The graphical view of the TinyPK protocol is given below.

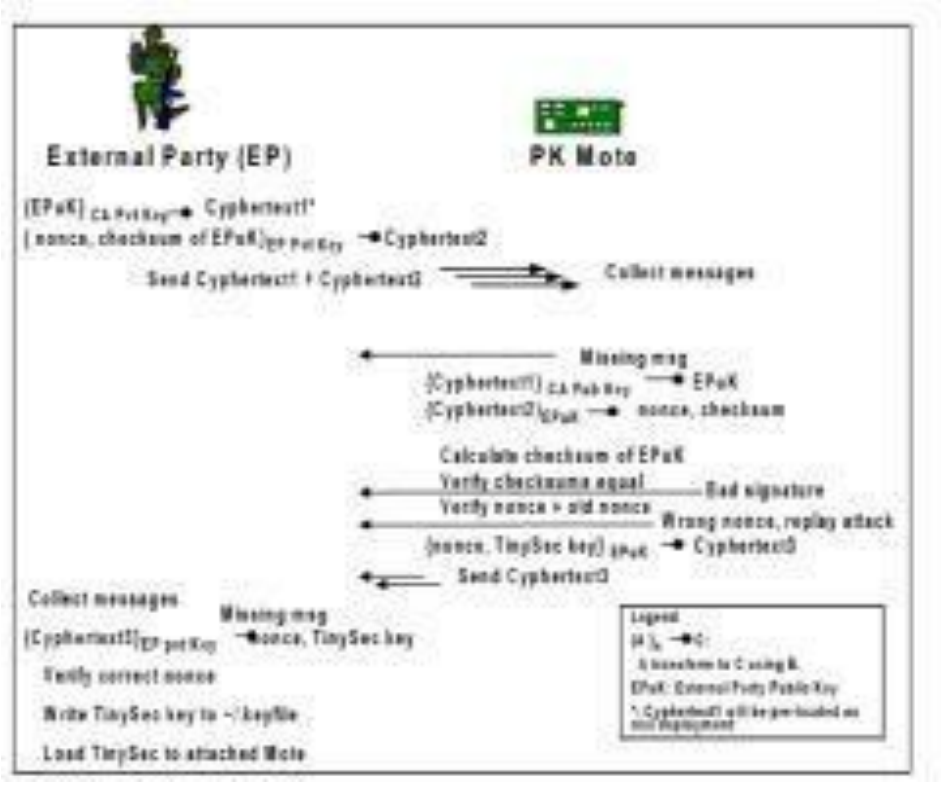

Fig 2. TinyPK EP Protocol Exchange Diagram

V. THE PROPOSED PKASK PROTOCOL

- provides Secrecy
- provides Authentication
- Prevents Node compromise attack
- Prevents Traffic Analysis Attack

BPubKey: Base station public key

Ebj: Secret key between Base station and Jth sensor

BPriKey: Base station Private Key The proposed protocol is based on RSA cryptosystem [11].The entities in the proposed protocol are Certification Authority (CA), External Agent (EA), Base station, Wireless Sensor networks which consists sensor nodes.

CA: A Certification Authority which is having private and public key pairs and trusted by sensors. The role of the CA in this process is to guarantee that the individual granted the unique certificate is in fact who he or she claims to be.

EA: External Agent is an entity who tries to communicate with the WSN. External agent also has a private and public key pair and the public key must be certified the CA.

Base station (BS): A Base Station is a signal transmitter or receiver that controls the local wireless network, and may also be the gateway between external environment and WSN. Base station plays a major role in WSN security. BS will have more computational and battery power compared to sensors.

Sensor nodes: A sensor (also called as mote) which is a constituent of WSN is capable of sensing, gathering, processing and communicating information to BS or EA.

Limitations of Tinypk

The main aim of public key cryptography is that the public keys of entities must be made public and private keys must be kept secret. In the above algorithm given by Watro [1], the public key CA and External Agent (EA) is not made public, which adds complexity in the protocol to send public key of External agent to sensor.

• If an intruder come to know the public key of CA which is made public then he can forge the system.

Communication between different entities of the proposed algorithm: Communication between External agent and Base Station: **RSA** based encryption and Decryption. Communication between Base station and Sensors: Secret key based encryption and decryption (e.g., AES/3DES).Communication between Sensor and External Agent as discussed in Watro et al [1] by using low key cryptography.

We explain the proposed protocol in three important phases. Registration, Authentication, and Session Key establishment phase.

We explain the proposed protocol in three important phases. Registration, Authentication, and Session Key establishment phase

A. Phase 1

First phase is registration. In registration phase the entities involved are External Agent and Base Station. The process involved is to register the External Agent public key with Base station.

- Any External Agent EA_i wants to communicate with a sensor must be certified by a CA in which the WSN trusts.
- On certified by CA, EA_i
- submits the public key $EPubKey_i$ to the base station.
- Base station validates the public key $EPubKey_i$ submitted by the External Agent EAi, by checking whether it is certified by the CA in which the network trusts.
- Once certified the Base station stores the public key ($EPubKey_i$), nonce, date of registration in its database as a record for the external agent EA_i.

B. Phase 2: Second phase is authentication. In Authentication phase the base station authenticates the external Agent.

- Any request COM-REQ which External agent wants to transfer to sensor, along with the nonce is encrypted with External Agent private key EPriKey .In the graphical view it is called as Ciphertext1.

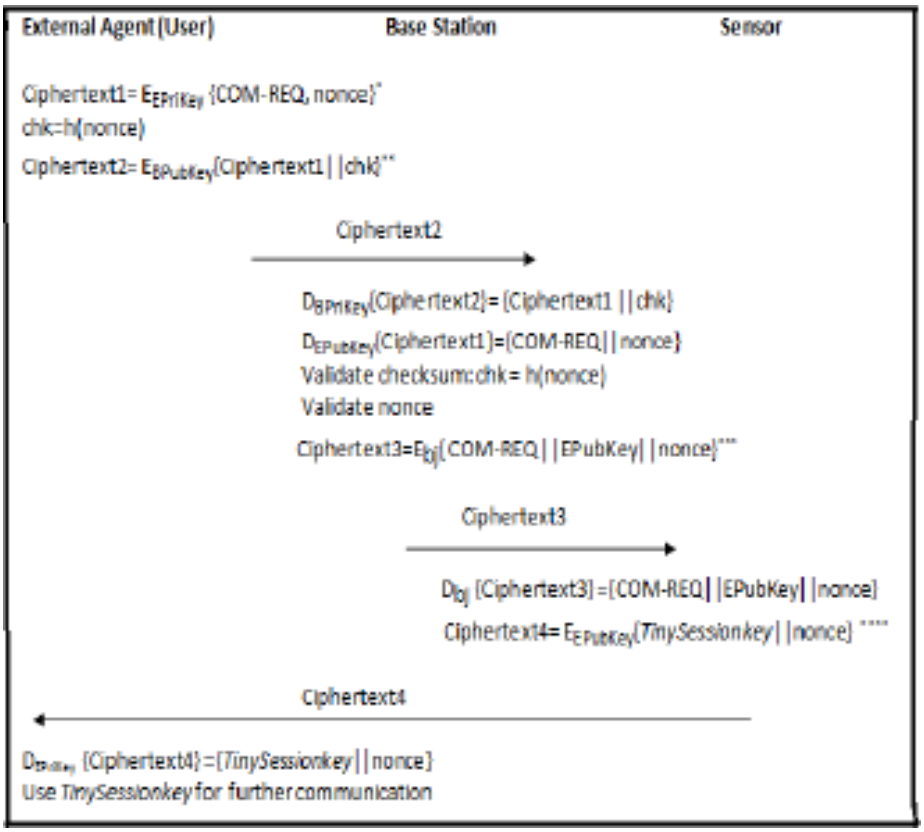

Fig 3.PKASK key Exchange diagram

Ciphertext1= $E_{EPrikey}$ {COM-REQ, nonce}.Encrypting the message with private key of EA, ensures that only EA could have sent it and provides Authentication of EA to the Base station. To check the integrity of Ciphertext1, checksum of nonce is calculated.chk=h(nonce).Ciphettext1 and checksum of nonce is encrypted with the public key of Base station(BPubKey)

i.e. $E_{BPubkey}$ {Ciphettext1||chk}which is called as Ciphertext2 in the graphical view. Encrypting the Ciphertext1 along with chk with the public key of base station ensures that only base station can decrypt it thus provided secrecy of message .Then the message is transferred over an insecure public channel.

- Base station decrypts the message (Ciphertext2) with its own private key($BPriKey$), that is $D_{BPrikey}$ {Ciphertext2}.Ondecrypting Ciphertext2 the base station gets {Ciphertext1||chk}.Base station decrypts the Ciphertext1 with the public key of External Agent ($EPubKey$). Once it decrypts Ciphertext1,that is $D_{\ EPubkey}$ {Ciphertext1} base station gets the COM-REQ and nonce. Base station validates the nonce by calculating the hash of it .If it is equal to chk, that is chk = h (nonce), then it proceeds to next step, else it rejects the request.
- Any valid request from any External agent EA_i , the base station updates the nonce in its record for that External agent.
- On Authentication of External agent request, the base station instructs a sensor node (say $J_{\ th}$ node) to respond against EA_i's request. Base station stores a secret key with each sensor node of that WSN. Base station computes Ciphertext3= E_{bj} {COM-REQ||EPubKey||nonce} and transfer the same to the $J_{\ th}$ sensor node.

- The J_{th} node decrypts Ciphertext3 i.e. D_{bj} {Ciphertext3} = {COM-REQ||EPubKey||nonce}.

*C. Phase 3 :*Third phase is establishment of session key.

- On getting COM REQ||EPubKey||nonce, The sensor frames a message Ciphertext4= $E_{EPubkey}$ $\{TinySessionkey,\ nonce\}$ As proposed in Watro [1] using of RSA algorithm to encrypt the message containing Session key with the public key of External agent assures only External agent can able to decrypt with its private key, which assures secrecy, which is an important factor while transmitting the session key.
- On getting the Ciphertext4 the External agent decrypts the message with its private key and validates the nonce. Once it is valid then the external agent communicates with the sensor through TinySessionkey.

VI. ANALYSIS OF PKASK PROTOCOL

In this section we show that the proposed protocol provides security in all the conditions mentioned in the section 3.Each possible attack on wireless sensor network and the steps taken the proposed algorithm is explained for each attack.

A. Eaves Dropping and Passive Monitoring:

This is most common and easiest form of attack on data privacy. If the messages are not protected by cryptographic mechanisms, then definitely the adversary can capture the message.

Steps taken to prevent this attack in the proposed protocol:

Every message that transmitted between External Agent and Base station, Base station and Sensor node, Sensor node and Base station is encrypted formats, Hence no scope for Eaves dropping and passive monitoring.

B.Traffic Analysis Attack:

Traffic Analysis is another common attack combined with Eaves dropping by the attackers. If two entities communicating frequently can denote those nodes have some specific activities and events to monitor. If the entities lack of communication denotes lack of activity between them.

Steps taken to prevent this attack in the proposed protocol:

Both the cases dealt carefully in the proposed algorithm, from the start of communication from External agent to Base station and transferring of Session key to

External agents are happening only once and in an encrypted format. Hence it doesn't give any clue to the adversary. (No mathematical operations like XOR etc. or not performed on the contents of the message, if so the adversary can easily analyses the contents of the message).

C.Replay Attack:

Replay attack is the one in which an adversary captures messages from an authorized user who is logging into a network and resent (replayed) to login as authorized user later.

Steps taken to prevent this attack in the proposed protocol:

In order to replay the valid request from External Agent, the adversary must know the nonce which is used by Base station as a distinguishing factor among requests. The adversary must need private key of base station to decrypt Ciphertext2 to get nonce, which is not possible .Hence by capturing Ciphertext2, the adversary won't gain anything.

D.Impersonation Attack (Phishing):

It is an attack in which an adversary confuses by imitating as legitimate External agent or a base station. The adversary has to capture the identity of an External agent to confuse a base station.

Steps taken to prevent this attack in the proposed protocol:

The adversary can confuse base station as a legitimate External agent, if it can get the nonce of the message (through which a base station distinguishes the external entities) then it can confuse Base station. As the

Ciphertext2 is in an encryption format and must be decrypted only private key of base station, hence it's not possible for an adversary to impersonate External Agent.

The entire message transferring between base station and sensor is done through a secret key, which is not possible for an adversary to get the secret key. In related to Impersonation attack, we discuss an algorithm provided by Maniklal das et al [15]. In which the session key from sensor to External agent is transmitted through XOR operations. With the simple example we prove that the protocol proposed by [15] is insecure.

RES1= COM-REQ $\oplus$ nonce {1}

RES2 = nonce $\oplus$ TinySessionKey {2}

RES1 and RES2 are transferred through insecure communication channel. Suppose if adversary got RES1 and RES2, he can perform following operations to get the session key from (2) we can deduct that RES2$\oplus$ nonce = TinySessionKey Assume nonce and Session key is of 4 bits. Adversary knows RES2 assume it is 1011. Substituting in the above (2), 1011$\oplus$ nonce = session key. For four bits, there are 16 possible values, 0000 to 1111 on each substitution of possible values from 0000 to 1111, the Adversary will send 16 messages to sensor, definitely among those 16 messages, and one message becomes valid which contains the valid session key and nonce as sent to external agent. The sensor assumes the adversary as trusty and may provide confidential information. So if we perform any mathematical operations to send the session key, the adversary can perform traffic analysis and can break the equations to get the session and nonce. The same applies to key of n bits. In our proposed protocol, the message containing Session key is encrypted with External agent public key with RSA algorithm. The message must be decrypted with the private key of External agent, which is not known to anyone .This makes our proposed protocol free from impersonate attack.

E.Node Compromise Attack:

Node compromise attack occurs when an attacker, though some subvert means gains control of the sensor node in the network after deployment. The adversary can simply extract information vital to the network's security such as data and security keys.

*Steps taken to prevent this attack in the proposed protocol:*Every external Agent or user trying to connect to WSN, the user authentication and data access must be done through Base station and after authentication, an encrypted message is sent to node to establish a session key for secure data access. The base station continuously monitors whether any node is captured or not. If a base station comes to know the compromised or malicious node, then the base station doesn't forward the message to that node. In TinyPk protocol, the external agent communicates to a sensor node directly, in these types of scenarios, node capturing is having higher probability compared to our proposed protocol.

F.Node colluding Attack:

Node colluding attack occurs when two nodes comes to an agreement to flood fraudulent messages in to the WSN.

*Steps taken to prevent this attack in the proposed protocol:*The base station maintains a unique secret key for each sensor node in WSN. All the messages to a sensor node from a base station are encrypted through the secret key meant for that sensor node. Of course this increases burden on the base station but to avoid Colluding attack and the insider attack it is the best possible solution. Hence in our protocol, if two nodes collude then it doesn't have any harm on the network.

VII. EFFICIENCY OF PKASK PROTOCOL

To analyze the efficiency of our proposed protocol, it is compared with TinyPk [1] in terms of computation cost and communication cost. It is quite evident that the proposed protocol is well suited for the resource–constrained WSN sensor node in comparisons to TinyPK protocol.

A. Computation Cost Analysis:

In the proposed protocol, the base station mainly performs two decryption activities. Provided with huge computational and Energy resources [12] Base station doesn't have any burden on implementing these encryption and decryption activities. In TinyPk a sensor with limited resources and energy made to perform three RSA based encryption and decryption operations compared to only one in the proposed protocol.

The notations used are:

TKPR : Computation time for a private key operation.

TKPU: Computation time for a public key operation.

TH : Computation time for a hash key operation.

TSK: Computation time for a secret key operation

Table 1. COMPARISON OF THE PROTOCOLS EFFICIENCY

Computation time -> Protocol	**External party**	**Base station**	**Sensor node**	**Result of Observation**
Tinypk	$2T_{KPR} + T_H$	*No*	$3T_{KPU} + T_H$	If an Adversary knows the public key of CA then protocol fails.
PKASK	$T_{KPR} + T_{KPU} + T_H$	$T_{KPR} + T_{KPU} + T_H + T_{Sk}$	$T_{Sk} + T_{KPU}$	It's not possible for an adversary in any way to break the protocol

Communication Cost Analysis:

Both TinyPK and our Proposed protocol consumes similar communication cost. As specified in [4], Communication is most costly than computation in WSN. In our proposed protocol we didn't increased the communication cost compared to Tiny PK.A sensor node in the WSN receives only one message from base station and transmits only one message to external agent. Once session key is established between External agent and the sensor node, the number messages exchanged between those two is dependent on the application. In terms of computation and Communication costs, our proposed protocol is efficient then the TinyPK protocol in terms of sensor node computation and energy usage. Based on the above comparison table we reduced the public key operation from three to one in our protocol.

VIII. CONCLUSION

We proposed a public key Cryptography based protocol for user Authentication and Session key establishment between external agent and a sensor in a secure manner. Based on Author knowledge this is first of its kind protocol which is most secure (No threat of forging an adversary in any means), low cost, user authentication and key establishment protocol. We compared our protocol with related protocols [1][15] and shown it is secure and efficient. This level of protection is achieved with very little overhead and has been shown to operate on the most limiting of sensor network platforms.

Our future work will study the problem of supporting mote networks that employ multiple session keys. As mote networks scale to larger sizes, the use of multiple session keys will be inevitable. Mote networks will need to internally generate new keys and deploy them as the communication patterns in the network change.

REFERENCES

[1] R. Watro, D. Kong, S. Cuti, C. Gardiner, C. Lynn, and P. Kruus, "TinyPK: Securing sensor networks with public key technology", In Proceedings of the 2nd ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN'04), pp. 59-64, New York, USA, 2004, ACM Press.

[2] A. Cerpa, J. Elson, D. Estrin, L. Girod, M. Hamilton, and J. Zhao, "*Habitat monitoring: Application Driver for Wireless Communications Technology*," 2001.

[3] V. A. Kottapalli, A. S. Kiremidjian, J. P. Lynch, E. Carryer, T. W. Kenny, K. H. Law, and Y. Lei, "*Two-tiered wireless sensor network architecture for structural health monitoring,*" SPIE's 10th Annual International Symposium on Smart Structures and Materials, March 2003.

[4] Vital Dust: Wireless Sensor Networks for Emergency Medical Care, http: //www.eecs.harvard.edu/»mdw/proj/vitaldust/.

[5] NEST Challenge Architecture, August 2002.

[6] A. Perrig, R. Szewczyk, V. Wen, D.E. Culler, and J.D. Tygar,"*SPINS: Security protocols for sensor networks*", Wireless *Networks*, Vol.8 , No. 5, pp. 521-534, September 2002.

[7] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D.E. Culler, and K. Pister, "*System architecture directions for networked sensors*",In Proceedings of the 9th International Conference on Architectural Support for Programming Languages and Operating Systems, New York, ACM Press, 2000, pp. 93-104.

[8] G.Gaubatz, J-P. Kaps, B. Sunar, "*Public Key Cryptography in Sensor Networks Revisited*", 1st European Workshop on Security in Ad-Hoc and Sensor Networks (ESAS 2004), Lecture Notes in Computer Science, vol. 3313, Springer, Heidelberg, pp. 2-18, August, 2004

[9] P.Jadia, A.Mathuria and Vanstone. Efficient Secure Aggregation in Sensor Networks. In: proc High performance Computing (HiPC), LNCS 3296, pp. 40-49, 2004**.**

[10] R.C.Merkle, "*Protocols for public key cryptosystems*", In Proceedings of the IEEE Symposium on Research in Security and Privacy, April 1980.International Journal of Communication Networks and Information Security (IJCNIS) Vol. 1, No. 2, August.

[11] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "*A survey on sensor networks*", *IEEE Communications Magazine*, Vol. 40, No. 8, pp. 102-114, August 2002.

[12] A.S. Wander, N. Gura, H. Eberle, V. Gupta, and S.C. Shantz,"*Energy analysis of public-key cryptography for wireless sensor networks*", In Proceedings of the 3rd IEEE International
Conference on Pervasive Computing and Communication, March 2005.

[13] D.W. Carman, P.S. Krus, and B.J. Matt, "*Constraints and approaches for distributed sensor network security*", Technical Report 00-010, NAI Labs, Network Associates Inc., Glenwood, MD, 2000.

[14] R.L Rivest,A.Shamir , and l.Adleman. *A method for obtaining digital signatures and public key crptosystems*.Communications of the ACM, Vol.21,No.2,pp.120-126 1978 .

[15] Vidhani Kumar and Manik Lal Das. *Securing Wireless Sensor Networks with Public Key Techniques*. Adhoc & Sensor Wireless Networks, pp.189-201, 2008**.**

BIBLIOGRAPHY OF AUTHORS

Swasthika Jain T J received the M.Tech degree in Computer Science and Engineering from Rajiv Gandhi Institute of Technology, Bengaluru. She is currently working as an Assistant Professor in Department of Computer Science Engineering, GITAM University, Bengaluru campus. Her research interests are in the networking field with focus on network security, wireless sensor network and internet of things.