RESEARCH ARTICLE OPEN ACCESS

Confidential DoS-Resistant Code Update Protocol in WSN

Deeksha S
(Student M.Tech, CSE, SIT, Tumkur
Email: s.deeksha45@gmail.com)

Abstract:

Wireless sensor network (WSN) consists of spatially distributed autonomous sensor nodes. WSNs are widely applicable in monitoring and control of environmental parameters such as temperatures, sound, pressure, etc. Sensor nodes cooperatively pass data among themselves in the network. Sometimes it is necessary to update data or program image through wireless links after they are deployed in order to adjust configuration parameters of sensors or distribute management commands and queries to the sensor nodes. In many applications these code updates need to be kept secret. Authentication and confidentiality play a very important role in providing secure code updates. Sometimes an adversary may transmit forged data, forcing nodes to waste their energy in identifying it as invalid. And also adversaries may sometimes repeatedly send invalid program image requests to its neighbours making them exhaust their energy. To avoid such attacks authentication of nodes need to be made and code updates need to be kept confidential.

Keywords— Wireless sensor networks, secure code updates, confidentiality, authentication, denial of service.

I. INTRODUCTION

Wireless sensor network (WSN) consist of collections of small, low-power nodes called sensor nodes. These sensor nodes are capable of performing some processing, gathering sensed information, communicating with other connected nodes in the network. Wireless sensor networks are used in various safety-critical applications such as critical infrastructure protection and surveillance, health monitoring, monitoring and control of environmental parameters etc. Sensor nodes cooperatively pass data among themselves through in the network. After deployment of sensor nodes in the network sometimes there is need to send code update data or new program image to update installed programs of stored parameters or to add new functionality or to perform software maintenance or to adjust configuration parameters or to distribute management commands and queries without having to physically reach each individual node in the network. These code update data is of the order of kilobytes while a parameter is just few bytes long. Due to such vast differences in their sizes, design of the dissemination protocols is Different. The two types of dissemination protocols are developed. Code dissemination protocols are developed to efficiently distribute long code update data into a network, enabling complete system reprogramming. On the other hand, short code updates data such as several two-byte configuration parameters, within a WSN. Common uses of this kind of protocols include injecting small programs, commands, queries, and configuration parameters. Sensor nodes are battery operated, so energy should be efficiently utilized. Security vulnerabilities occur in the network, which allow adversary to update undesirable values, erase critical values or launch DoS (Daniel of Service) attack in the network. Lack of authentication may allow attacker to attack one sensor node and misuse it to send false program image into the network and take control over entire network. And in some applications like in military, environmental applications these code updates need to be kept secret because adversaries may gain access to these code update information and misuse it. In many code update protocols Digital signatures have been used, but power consumption is more, as sensor nodes are battery operated energy should be efficiently utilized. Digital signature is used for authentication in code dissemination protocols, an adversary can repeatedly broadcast false data packet with an invalid signature to sensor node making them exhaust their energy by causing unnecessary signature verifications. And in multi-hop code dissemination protocols, sensor nodes need to broadcast their new program image to their neighbors in response to a request. Hence, an adversary can repeatedly send request to same program image to its neighbors and make them rebroadcast the same program image, until the power resources of the sending node depletes.

II. RELATED WORK

Deluge [1] is a quick, reliable data dissemination protocol. In Deluge large amount of data can be propagated from one or more sensor nodes to all other nodes over a multi hop, wireless sensor network. In Deluge code update program is divided into fixed sized blocks called pages. And these pages are divided into fixed sized packets. Deluge employs three phase process - Advertisement phase, Request phase, Update phase. In Deluge each node periodically broadcast the program version available for sending. This advertisement is not authenticated, and any node in the network may advertise any version number. Deluge also exports a simple, one-hop network querying interface. Any node in the network which is not authenticated can query to obtain the program version number currently installed. Without authenticated broadcasts an attacker can advertise a new version number, can broadcast false data or program to any number of nodes, and can hijack the entire network and can take control over the network. Deluge uses 16-bit cyclic redundancy checks (CRC) across both packets and pages to verify program integrity. The CRCs do not protect against an attacker injecting a bogus program or program fragment.

To provide security to Deluge network programming [2], hash function is used. Hash of each packet is computed and placed in the previous packet. The hash of the first packet is digitally signed using private key of Base Station. And each node verifies the first packet with Base Station's public key. Seluge [3] is an efficient, secure, robust, and DoS-resistant code dissemination system for wireless sensor networks. Seluge relies on Deluge for efficiency and robustness. Seluge further provides three layers of protection: Immediate authentication of code dissemination packets, authentication of page advertisement and SNACK packets, and anti-DoS protection for signature packets.

III. DESIGN

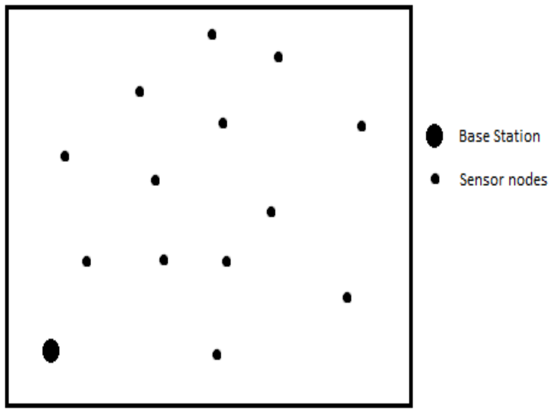

Fig1: BS and Sensor nodes deployment region

The network area consists of Base Station and sensor nodes. BS will have public and private key pair. And all sensor nodes in the network will be pre-configure with public key of BS. The code dissemination scheme consists of 3 phases:

1. Initialization phase
2. Packet processing phase
3. Verification phase

1. Initialization phase:

BS will generate one-way hash chain of length 'n' by applying hash to an initial element 'n' times. The result obtained is called committed value. The hash values are used as cryptographic keys. The length of hash chain must be at least equal to the number of expected program image updates. The ith element in this hash chain is denoted as V_i.
Then we have

$$V_i = H(V_{i+1}),\ 0<=i<=n.$$

Where: n=maximum number of program image updates.

The committed value (V_0) will be pre-loaded in each node before deployment along with Base station's public key. The ith element of this hash chain will be stored in sensor node is called version key.

2. Packet processing phase:

After initialization phase, if BS has any program image updates then it will prepare the data packets. Initially, an M-byte nonce (i.e., Q) is concatenated with the last packets of the program image (i.e., P_n). Then hash function is applied to $P_n||Q$ i.e.,

$$h_n <= H(P_n||Q).$$

The first M bytes of h_n are used as a session key to encrypt Pn||Q, which produces last encrypted data packet ($E_{hn}(P_n||Q)$). Next, the first M bytes of h_n are concatenated with the previous packet (P_{n-1}). Again, the one-way hash function is applied to $P_{n-1}||h_n$, yielding h_{n-1}. The encryption process for P_{n-1} is similar to that for P_n. This concatenation-hashing-encryption process is applied to all program image data packets until the first packet of the whole program image is reached. After h_1 is used to encrypt first packet ($P_1||h_2$), it will be used in the module called cipher puzzle constructor, h_1 is digitally signed by the Base station ($Sig(h_1)$).

A weak authenticator called Cipher Puzzle (CP) is constructed by finding fixed size segments(S) which satisfies

$$ETM_{V0}(h_1|S|V_1) = H(Sig(h_1))$$

in the first k bytes, where k is the strength of the Cipher Puzzle. Cipher Puzzle is used to defend against a signature-based attack. $Sig(h_1)$ and the cipher text $ETM_{V0}(h_1|S|V_1)$ without first k bytes, are included in Advertisement packet, which will be broadcasted initially to sensor nodes to alert sensor nodes about the new update that is available. When new program image is to send afte ith updates, the above procedure is repeated using nonce for the last data packets different from all I previous updates.

3. Verification phase:

This phase comprises of three stage process i.e., Advertisement verification, Request verification and Update verification.

3.1 Advertisement verification:

Each advertisement verification packet contains a number called version number of the program image update. Each node after receiving advertisement packet will check the version number and check whether it is a newer than the current version. And to avoid vulnerabilities node further investigates by checking whether decrypted V_{i+1} and the current version key are same. In addition Request_counter is set to 0 to avoid Request-Based Dos attack during request verification i.e., Request_counter will be used to check how many times same request has been made by the neighboring node.

3.2 Request verification:

After verifying advertisement packet, node encrypts the puzzle solution and the sequence number updated of the same program image. This encrypted result is called request authenticator is attached to the Request packet and will be unicasted to the sender. The sending node will decrypt the authenticator packet and will initialize the respective sequence number for the receiver. And at the same time sending node will check whether same request has been made more than threshold value (i.e., Max_updates). If this request has not been made more that Max_updates times then the request will be processed and Request_counter will be incremented by 1.

3.3 Update verification:

When the request is accepted sending node will prepare the update packets and will send it to its one-hop neighbors. Receiving nodes after receiving encrypted packets, it uses session key from (i+1)th packet to decrypt ith packet. After decryption one-way hash relationship between decrypted packets and session key is checked. This check is to ensure the integrity of update packets.

IV. CONCLUSIONS

In this work, a new security scheme is proposed to provide confidentiality and Dos-resistance in muti-hop code update in WSN. Here session keys are derived from hashing data packets inorder to encrypt the same data packets. And Cipher Puzzle is used as a weak authenticator to provide confidentiality for the first session key which is used to defend against signature-based and request-based DoS attack.

REFERENCES

1. *Jonathan W. Hui and David Culler. The Dynamic Behavior of a Data Dissemination Protocol for Network Programming at Scale,In SenSys '04, pages 81-94, Nwe York, NY, USA, 2004, ACM press.*

2. *Dutta Prabal K, Hui Jonathan W, Chu David C, Culler David E. " Securing the deluge network programming system ". In: IPSN'06: Processing of International Conference on Information Processing in Sensor Networks. New York, NY, USA: ACM Press; 2006.p.326-33.*

3. *Hyun, S.; Ning, P.; Liu, A.; Du, W. Seluge: Secure and DoS-Resistant Code Dissemination in Wireless Sensor Networks. In Proceedings of 2008 International Conference on Information Processing in Sensor Networks, St. Louis, MO, 22–24 April 2008; pp. 445–456.*

4. *Ning Peng, Liu An, Du Wenliang. DoS attacks against broadcast authentication in wireless sensor network. ACM Transaction on Sensor Networks January 2008;4(1):1-35.*

5. *Lanigan PE, Gandhi R, Narasimhan P. Sluice: Secure dissemination of code updates in sensor networks. In: 26th IEEE International Conference on Distributed omputing Systems, 2006(ICDCS 2006); 2006. p. 53-63.*

6. *Hyun Sangwon, Ning Peng, Liu An. Mitigating wireless jamming attacks via channel migration. In: Distributed Computing Systems Workshops (ICDCSW), 2011 31st International Conference on. IEEE; June 2011. p. 313-22.*

7. *Hu YC, Perrig A, Johnson DB. Packet leashes: a defense against wormhole attacks in wireless networks. In: INFOCOM '03:Proceedings of 23rd IEEE International Conference on Computer Communications, vol. 3; 2003. p. 1976-86.*

8. *Hu Wen, Tan Hailun, Corke Peter, Shih Wen C, Jha Sanjay. Towards trusted wireless sensor networks, vol. 7.New York, NY, USA: ACM; August 2010. 1-25.*

9. *Hu Wen, Chou Chun T, Jha Sanjay, Bulusu Nirupama. Deploying long-lived and cost-effective hybrid sensor networks. Ad Hoc Networks Nov 2006;4(6):749-67.*

10. *Du X, Guizani M, Xiao Y, Chen HH. Defending DoS attacks on broadcast authentication in wireless sensor networks. In:Communications, 2008. ICC '08. IEEE International Conference ; 2008.p.1653-7.*

11. *Deng Jing, Han Richard, Mishra Shivakant. Secure code distribution in dynamically programmable wireless sensor networks. In: IPSN'06:Processings of International Conference on Information Processing in Sensor Networks. New York, NY, USA: ACM Press; 2006.p. 292-300.*

12. *Jeong Jaein, Culler D. Incremental network programming for wireless sensors. In: Sensor and Ad Hoc Communications and Networks, 2004. IEEE SECON 2004. 2004 First Annual IEEE Communications Soceity Conference on; 2004.p. 25-33.*